

Общество с ограниченной ответственностью
«СВД Встраиваемые Системы» (ООО «СВД ВС»)
ул. Кузнецовская, д. 19, г. Санкт-Петербург, 196128

тел. (812) 346-89-56, факс (812) 346-89-53
www.kpda.ru sales@kpda.ru

ПК ЦКИ

1.4

Программный комплекс поддержки работы с цифровой картографической
информацией

30 апреля 2025 г.

Содержание

ПК ЦКИ	2
Содержание	2
ПК ЦКИ для ЗОСРВ «Нейтрино»	10
Инструкция по установке	11
Состав комплекса	11
Установка и запуск компонентов среды исполнения	11
Установка компонентов среды разработки	12
Сборка демонстрационных приложений	12
Документация	13
Контакты	13
Руководство разработчика	14
Обзор программного комплекса	16
Работа с движком визуализации	18
Параметры отображения	18
Пользовательские объекты	22
Преобразования и измерения	22
Запуск и отображение	23
Переменные окружения	26
Общие переменные окружения	26
Переменные окружения рендеринга	26
Описание API	28
Управление отладочным выводом	57
gis_helper_debug_file_getpath()	58
gis_helper_debug_file_setname()	60
gis_helper_debug_mode_getmask()	62
gis_helper_debug_mode_name()	63
gis_helper_debug_mode_setlvl()	65
gis_helper_debug_mode_setmask()	67
gis_helper_debug_write()	69
gis_helper_debug_write_lvl()	71
gis_helper_debug_write_mask()	74
gis_debug_level_t	76
gis_debug_mode_t	77
Управление картографическим ядром (сервисом)	77
gis_core_class_data_base_code()	78
gis_core_class_data_init()	80
gis_core_class_list_free()	82
gis_core_class_list_init()	83
gis_core_class_list_reinit()	84

gis_core_link_connect()	85
gis_core_link_destroy()	87
gis_core_link_init()	89
gis_core_map_list_free()	91
gis_core_map_list_get_entry()	92
gis_core_map_list_init()	94
gis_core_request_connection_state()	95
gis_core_request_download_maps()	97
gis_core_request_driver_acronym()	99
gis_core_request_driver_connect()	101
gis_core_request_driver_info()	103
gis_core_request_drivers_prefix()	105
gis_core_request_map_class_list()	107
gis_core_request_maps_list()	109
gis_core_request_parameters_are_correct()	112
gis_core_request_parameters_init()	114
gis_core_request_parameters_set_borders()	115
gis_core_request_parameters_set_data_source()	117
gis_core_request_parameters_set_driver()	119
gis_core_request_revision()	121
gis_core_request_update_cache()	123
gis_core_request_update_cached_map()	125
gis_helper_get_core_data_source_name()	127
gis_helper_get_map_source_by_driver_id()	129
gis_helper_is_data_source_valid()	131
gis_core_class_code_t	133
gis_core_class_data_t	134
gis_core_class_info_t	135
gis_core_class_list_t	136
gis_core_connection_state_t	137
gis_core_connection_t	138
gis_core_driver_id_t	139
gis_core_driver_info_t	141
gis_core_hash_t	142
gis_core_map_data_source_t	143
gis_core_map_information_t	145
gis_core_map_list_t	148
gis_core_request_parameters_t	149
gis_core_update_cache_mode_t	150
GIS_CORE_DRIVER_FOR_LOOP_HEAD	152
GIS_CORE_DRIVER_ID_CHECK	154
GIS_IS_CORE_MAP_DATA_SOURCE_VECTOR	156
Управление стилем отображения векторных карт	157
gis_helper_get_map_style_by_data_source()	158
gis_helper_get_map_style_name()	160
gis_helper_is_map_style_valid()	162
gis_map_style_fill_active_classifier_name_list()	164
gis_map_style_flip_language()	166
gis_map_style_get_active_palette_idx()	168
gis_map_style_get_palette_list()	170
gis_map_style_get_SEABED()	172

gis_map_style_palette_get_entry_count()	175
gis_map_style_palette_get_entry_list()	177
gis_map_style_palette_get_map_style()	179
gis_map_style_palette_list_free()	181
gis_map_style_set_active_palette_idx()	183
gis_map_style_set_SEABED()	185
gis_map_style_t	187
gis_vector_palette_list_t	188
gis_vector_palette_t	189
Управление областью данных	189
gis_core_databuffer_attach()	190
gis_core_databuffer_check()	192
gis_core_databuffer_data_request()	194
gis_core_databuffer_detach()	196
gis_core_databuffer_resize()	198
gis_core_databuffer_size()	200
gis_object_free()	202
gis_object_init()	204
gis_databuffer_desc_t	206
class ClassifierInfo	207
class ClassifierLayerInfo	208
class DBCallbacks	209
class LayerInfo	210
class MapInfo	211
class SMCallbacks	212
DBCallbacks::databufferValidityChangedCallback()	213
SMCallbacks::errorCallback()	215
SMCallbacks::mapClassRequiredCallback()	217
SMCallbacks::mapLayerCompleteCallback()	219
SMCallbacks::mapLayerRequiredCallback()	220
SMCallbacks::surfaceCompleteCallback()	221
gis_object_t	222
class GisObjectList	224
GisObjectList::find_nearest_object()	225
GisObjectList::find_nearest_point()	227
GisObjectList::get_object()	229
GisObjectList::get_object_count()	231
GisObjectList::GisObjectList()	232
GisObjectList::GisObjectList()	233
Управление движком визуализации (рендеринга)	234
gis_data_engine_alloc()	235
gis_data_engine_free()	237
gis_data_engine_get_canvas_size()	239
gis_data_engine_get_classifier_list()	242
gis_data_engine_get_class_list()	245
gis_data_engine_get_display_parameters()	248
gis_data_engine_get_map_list()	250
gis_data_engine_get_maps_projection()	253
gis_data_engine_set_canvas_size()	256
gis_data_engine_set_notify_func()	258
gis_data_engine_update()	260

gis_data_engine_validity_changed()	262
gis_data_engine_context_t	264
Параметры отображения карты	264
gis_gui_get_projection_parameters()	265
gis_gui_get_screen_parameters()	268
gis_map_ellipsoid_get_full_name()	270
gis_map_height_system_get_full_name()	272
gis_map_projection_get_full_name()	274
gis_map_projection_has_zone()	276
gis_map_projection_init()	278
gis_map_projection_is_filled()	280
gis_map_projection_zero()	282
gis_mdp_get_background_color()	284
gis_mdp_get_brightness_contrast()	286
gis_mdp_get_center_point_deg()	289
gis_mdp_get_display_resolution()	291
gis_mdp_get_display_size()	293
gis_mdp_get_map_antialiasing_level()	295
gis_mdp_get_phys_scale()	297
gis_mdp_get_projection()	299
gis_mdp_get_raster_color_mode()	301
gis_mdp_get_raster_height_mode()	303
gis_mdp_get_raster_invalid_height_color()	305
gis_mdp_get_raster_lower_height_limit()	307
gis_mdp_get_raster_palette()	309
gis_mdp_get_raster_upper_height_limit()	312
gis_mdp_get_scaling_mode()	314
gis_mdp_set_background_color()	316
gis_mdp_set_brightness_contrast()	318
gis_mdp_set_center_point()	321
gis_mdp_set_center_point_ptr()	323
gis_mdp_set_display_resolution()	325
gis_mdp_set_display_size()	327
gis_mdp_set_map_antialiasing_level()	329
gis_mdp_set_phys_scale()	331
gis_mdp_set_projection()	333
gis_mdp_set_raster_color_mode()	335
gis_mdp_set_raster_height_limits()	337
gis_mdp_set_raster_height_mode()	339
gis_mdp_set_raster_invalid_height_color()	341
gis_mdp_set_raster_palette()	343
gis_mdp_set_scaling_mode()	346
gis_map_ellipsoid_idx_t	348
gis_map_height_system_idx_t	349
gis_map_projection_idx_t	351
gis_map_projection_t	352
gis_mdp_t	354
Движок surfacemanager	354
gis_render_sm_alloc()	355
gis_render_sm_calculate_distance()	357
gis_render_sm_calculate_polygon()	360

gis_render_sm_convert_degree2fpx()	363
gis_render_sm_convert_degree2px()	366
gis_render_sm_convert_fpx2degree()	368
gis_render_sm_convert_fpx2meters()	371
gis_render_sm_convert_px2degree()	374
gis_render_sm_convert_px2meters()	377
gis_render_sm_draw()	380
gis_render_sm_free()	382
gis_render_sm_get_draw_canvas_borders()	384
gis_render_sm_get_layer_type()	386
gis_render_sm_get_render_mode()	388
gis_render_sm_get_view_pixmap()	390
gis_render_sm_move()	392
gis_render_sm_redraw_userobject()	394
gis_render_sm_rescale()	397
gis_render_sm_set_layer_data_sources()	399
gis_render_sm_set_layer_type()	402
gis_render_sm_set_render_mode()	404
gis_render_sm_set_update_func()	406
gis_render_sm_sync()	408
gis_render_sm_update()	410
gis_render_sm_userdata_add_bitmap()	412
gis_render_sm_userdata_add_contour_object()	414
gis_render_sm_userdata_add_polygon()	417
gis_render_sm_userdata_add_polyline()	420
gis_render_sm_userdata_delete_contour_object()	423
gis_render_sm_userdata_delete_object()	426
gis_render_sm_userdata_edit_object_color()	429
gis_render_sm_userdata_edit_object_points()	432
gis_render_sm_context_t	435
gis_render_sm_layer_type_t	436
gis_userobject_t	437
Движок renderbuffer	437
gis_render_rb_alloc()	438
gis_render_rb_draw()	440
gis_render_rb_free()	442
gis_render_rb_context_t	444
Общие функции и типы данных	444
gis_data_raw_ctx_alloc()	445
gis_data_raw_ctx_free()	447
gis_data_raw_ctx_import_databuffer()	448
gis_data_raw_get_map_count()	450
gis_data_raw_get_raster_block_count()	452
gis_data_raw_get_raster_block_height()	455
gis_data_raw_get_raster_block_upper_left_corner()	458
gis_data_raw_get_raster_block_width()	461
gis_data_raw_get_raster_data()	464
gis_data_raw_get_raster_height_in_blocks()	467
gis_data_raw_get_raster_height_in_elements()	470
gis_data_raw_get_raster_meters_per_element()	473
gis_data_raw_get_raster_upper_left_corner()	476

gis_data_raw_get_raster_width_in_blocks()	479
gis_data_raw_get_raster_width_in_elements()	482
gis_data_raw_map_get_borders()	485
gis_data_raw_map_get_data_source()	487
gis_data_raw_map_get_filename()	489
gis_data_raw_map_get_object_count()	491
gis_data_raw_map_get_projection()	493
gis_data_raw_map_select_first_object()	495
gis_data_raw_map_select_next_object()	498
gis_data_raw_map_select_object()	501
gis_data_raw_object_get_bounding_rect()	504
gis_data_raw_object_get_class_code()	507
gis_data_raw_object_get_data()	509
gis_data_raw_object_get_height()	512
gis_data_raw_object_get_point_count()	514
gis_data_raw_object_get_type()	516
gis_data_raw_raster_maps_get_height_limits()	519
gis_data_raw_select_first_map()	522
gis_data_raw_select_map()	524
gis_data_raw_select_map_by_index()	527
gis_data_raw_select_next_map()	529
gis_gui_about()	531
gis_helper_are_maps_equal()	534
gis_helper_convert_point_degrees_2_meters()	536
gis_helper_convert_point_meters_2_degrees()	538
gis_helper_convert_point_meters_2_degrees_array()	540
gis_helper_env_get_config_directory()	543
gis_helper_env_get_gis_root_directory()	544
gis_helper_env_get_maps_cache_directory()	545
gis_helper_free_string_list()	547
gis_helper_math_free_ctx()	549
gis_helper_math_generate_meters_projection()	551
gis_helper_math_get_degrees_projection()	553
gis_helper_math_get_meters_projection()	555
gis_helper_math_init_ctx()	557
gis_helper_split_string_by_delimiter()	559
gis_object_primitive_type_get_full_name()	561
gis_borders_t	563
gis_data_raw_context_t	564
gis_helper_math_ctx_t	565
gis_object_primitive_type_t	566
int32_point_t, float_point_t, double_point_t	567
object_point_t	568
CLIP_VALUE	569
DEGREES_TO_RADS	571
EQUAL_POINTS_XY	572
EQUAL_POINTS_XY_PTRS	574
GETSIGN	576
RADS_TO_DEGREES	578
Дополнительные API	578
API для написания пользовательского драйвера форматов карт	578

gis_common_driver_check_folders	579
gis_common_driver_fill_extensions()	581
gis_common_driver_get_basename_without_extension	583
gis_common_driver_get_mandatory_driver_param_from_cfg()	585
gis_common_driver_recover_gcm_database()	587
gis_common_driver_reset_gcm_database()	589
gis_common_driver_sync_cache()	591
gis_common_driver_update_cache()	593
gis_common_driver_update_clipped_map_info()	595
gis_common_driver_extension_list_t	597
gis_driver_ctx_t	598
gis_driver_interface_t	600
GIS_CACHE_SUBDIR_GCM	602
GIS_GCM_FILE_EXTENSION	603
Работа с конфигурационными файлами	603
gis_core_config_get_driver_value()	604
gis_core_config_get_value()	606
gis_core_config_set_driver_value()	608
DRIVER_CFG_*, DRIVER_*_CFG_SECTION	610
Работа с таймером	611
gis_helper_timer_get_result_sec()	612
gis_helper_timer_start()	614
gis_helper_timer_stop()	616
gis_timer_t	618
Библиотека kd-tree	618
gis_kd_add()	619
gis_kd_destroy()	621
gis_kd_dnn()	622
gis_kd_init()	624
gis_kd_knn()	625
gis_kd_rnn()	627
gis_kd_node_t	629
gis_kd_tree_t	630
gis_kd_uniq_id	631
Работа с объектами	632
Разработка драйверов источников данных	635
Описание доступных драйверов	635
Особенности сборки драйверов	636
Особенности проектирования драйверов	636
Структуры и макросы	636
Структура gis_driver_ctx_t	637
Структура gis_driver_interface_t	637
Функции	639
Прототипы требуемых функций	639
Примеры реализации требуемых функций	639
Функция bmp_local_init	639
Функция bmp_local_convert_map	643
Функция bmp_local_connect	644
Функция bmp_local_get_connection_state	645
Функция bmp_local_get_info	645
Функция bmp_local_reset_file_cache	645

Функция bmp_local_synchronize_file_cache	645
Функция bmp_local_update_file_cache	646
Безопасность использования	647
Справочник по утилитам	649
Приложения	651
Инструменты контроля и мониторинга	651
gis-monitor	652
Средства визуализации	652
gis-filter-generator	653
gis-map-linker	654
gis-map-viewer	655
gis-raster-preview	657
gis-rb-viewer	658
Системные компоненты	660
Перечень системных компонентов	660
gis-core	661
gis-raster-processor	664
gis-s57-processor	666
gis-shape-processor	667
gis-sxf-processor	669
Утилиты	671
Утилиты общего назначения	672
gis-buffer-renderer	673
gis-control	675
gis-gcm-clip	677
gis-gcm-info	678
gis-mapstyle-info	679
Утилиты для работы с картами S-63, S-57	680
gis-s63-decrypt	681
gis-s63-gen-userpermit	683
gis-s63-unpack	684
Утилиты для работы с растровыми данными	684
gis-rasterizer	685

ПК ЦКИ для ЗОСРВ «Нейтрино»

Программный комплекс поддержки работы с цифровой картографической информацией



Актуальная версия: 1.4

Документ	Аннотация
Инструкция по установке	Инструкция содержит указания по установке компонентов ПК ЦКИ на целевую и инструментальную системы
Руководство разработчика	Описываются подходы к использованию компонентов программного комплекса поддержки работы с цифровой картографической информацией
Справочник по утилитам	Описываются информационные, диагностические и утилитарные компоненты комплекса и способы их использования

Техническая поддержка

Для получения технической поддержки напишите на электронный адрес support@kpda.ru или обратитесь по телефону (812) 346-89-56.

Уведомление об авторских правах

© 2017-2025, ООО «СВД ВС».

Для получения дополнительной информации обратитесь к [разработчику](#).

Инструкция по установке

Инструкция содержит указания по установке компонентов ПК ЦКИ на целевую и инструментальную системы



Полное название продукта: Программный комплекс поддержки работы с цифровой картографической информацией «ПК ЦКИ»

Сокращенное название: ПК ЦКИ

Альтернативные названия: Программное изделие КПДА.14903-01, Картографический пакет «ПК ЦКИ»

- Состав комплекса
- Установка и запуск компонентов среды исполнения
- Установка компонентов среды разработки
- Сборка демонстрационных приложений
- Документация
- Контакты

Состав комплекса

`"kpda-gis*.tar.xz"`

Архив с компонентами среды исполнения ПК ЦКИ. Архив предназначен для установки на целевую систему.

`"kpda-gis*-dev*.tar.xz"`

Архив с компонентами среды разработки ПК ЦКИ. Архив предназначен для установки на инструментальную систему. Архив в том числе содержит электронную документацию ПК ЦКИ.

Установка и запуск компонентов среды исполнения

Пример установки на целевую систему архитектуры x86 с установленной ЗОСРВ «Нейтрино» ред. 2018. Поместить архив с компонентами среды исполнения в корень ФС целевой системы:

```
# mv /path/to/archive/kpda-gis*.tar.xz /
```

Распаковать архив от корня:

```
# cd /  
# tar -Jxvf kpda-gis*.tar.xz
```

При необходимости установить архив с набором демонстрационных карт, доступный по запросу. Архив с набором демонстрационных карт поместить в корень ФС целевой системы:

```
# mv /path/to/archive/gis-*-demo-maps.tar.xz /
```

Распаковать архив от корня:

```
# cd /  
# tar -Jxvf gis-*-demo-maps.tar.xz
```

Установить права на исполнение компонентов ПК ЦКИ:

```
# chmod -R +x /opt/gis
```

Установить переменные окружения:

```
# export PATH=/opt/gis/bin:/opt/gis/sbin:$PATH  
# export LD_LIBRARY_PATH=/opt/gis/lib:$LD_LIBRARY_PATH  
# export ABLANG=ru_RU
```

Запустить [картографический сервис](#) с необходимым драйвером (например, *SXF*) и опцией **sync**, запускающей конвертацию карт по необходимости. Полный перечень доступных драйверов и опций запуска описан в документации.

```
# gis-core -dsxf-local,sync &
```

Запустить приложение [gis-monitor](#), позволяющее просматривать общую информацию о доступных картах:

```
# gis-monitor &
```

Запустить средство просмотра [gis-map-viewer](#) в районе города Подольск (при наличии набора демонстрационных карт):

```
# gis-map-viewer -x 37.787 -y 55.4914 -w 800 -h 600 -s 8 &
```

В результате появится окно Средство просмотра ЦКИ с картой Подольска (SXF).

Установка компонентов среды разработки

Установка выполняется на 64-разрядную инструментальную систему под управлением GNU/Linux или Windows с установленным Комплексом разработчика для ЗОСРВ «Нейтрино».

Скопировать архив с компонентами среды разработки на инструментальную машину. Например, в `/home/user/sandbox` для GNU/Linux или в `C:\sandbox` для Windows.

На инструментальной машине распаковать архив `kpda-gis*-dev*.tar.xz`, например в GNU/Linux:

```
# cd /home/user/sandbox  
# tar xvf kpda-gis*-dev*.tar.xz
```

В Windows распаковать архив `kpda-gis*-dev*.tar.xz` с помощью программ *7-Zip* или *WinRAR*.



Распаковка должна выполняться с правами администратора.

В результате распаковки в рабочей директории появится каталог `gis`.

Сборка демонстрационных приложений

Для сборки приложения [gis-monitor](#) необходимо:

- Перейти в каталог `gis/src/apps/gis-monitor`
- Определить переменную окружения `GIS_INSTALL_ROOT`, которая должна содержать абсолютный путь к директории `gis`, распакованной из архива на предыдущем шаге.
- Выполнить сборку под требуемую архитектуру.

Например, сборка для целевой системы x86 в GNU/Linux:

```
# cd /home/user/sandbox
# cd gis/src/apps/gis-monitor
# export GIS_INSTALL_ROOT=/home/user/sandbox/gis
# CPULIST=x86 make install
```

Аналогичная сборка в Windows:

```
# cd C:\sandbox
# cd gis/src/apps/gis-monitor
# set GIS_INSTALL_ROOT=C:\sandbox\gis
# set CPULIST=x86
# make install
```

В результате приложение [gis-monitor](#) соберётся и установится в каталог `gis/bin/nto/x86/opt/gis/bin`.

Сборка приложения [gis-map-viewer](#) (`gis/src/apps/gis-map-viewer`) выполняется аналогичным образом.

Для проверки перенести собранные программы [gis-monitor](#) и [gis-map-viewer](#) на целевую систему и запустить (см. [Установка и запуск компонентов среды исполнения](#)).

Документация

Онлайн документация для ПК ЦКИ доступна по ссылке: <http://gis.help.kpda.ru/help/index.jsp>

Для интеграции электронной документации в *Momentics IDE* необходимо:

1. распаковать файл `doc.gis*.tar.gz` из каталога `gis/doc`.
2. полученный jar-файл скопировать в каталог `ide/plugins`, внутри каталога, в который был установлен комплект разработчика.
3. запустить *Momentics IDE*.

Если при запуске *Momentics IDE* пакет документации не появится, то необходимо очистить индексированный список плагинов:

```
# rm -Rf ide/configuration/org.eclipse.help.base/index/*
```

Контакты

ООО «СВД ВС»

196128, г. Санкт-Петербург, ул. Кузнецовская, д. 19

Е-mail : support@kpda.ru

Тел. : (812) 346-89-56

Факс : (812) 346-89-53

Руководство разработчика

Описываются подходы к использованию компонентов программного комплекса поддержки работы с цифровой картографической информацией

Руководство разработчика предназначается для разработчиков картографических приложений. Оно содержит как рекомендованные практики программирования, так и описание предлагаемых API. В нем описано как создавать картографические приложения, используя библиотеки *gishelper* и *gisrender*.

Следующая таблица содержит информацию, которая поможет найти интересующую информацию:

Ссылка	Раздел документации
Обзор программного комплекса	Обзор программного комплекса поддержки работы с цифровой картографической информацией и общие рекомендации по применению на практике
Переменные окружения	Используемые переменные окружения и их краткое описание
Описание API	Описание API включает в себя группировку используемых функций и типов
Работа с объектами	Поиск объектов, получение информации о них, а также их выделение
Разработка драйверов источников данных	Разработка собственных драйверов источников данных на примере растрового формата BMP
Безопасность использования	Дается представление об особенностях использования функций системных библиотек

Типографические соглашения

Мы используем значки "заметки", "предостережения", и "предупреждения" для выделения важных сообщений:



"Заметки" указывают на нечто важное или полезное.



"Предостережения" сообщают о командах или процедурах, которые могут иметь нежелательные или побочные эффекты.



"Предупреждения" сообщают о командах или процедурах, которые могут быть опасны для Ваших файлов, данных, оборудования или даже Вас лично.

Вниманию пользователей Windows

В этой документации используется символ "прямой слэш" (/) в качестве разделителя во всех путях поиска, включая те, которые относятся к файлам в Windows.

Также в большинстве случаев мы следуем соглашениям о файловых системах в POSIX/UNIX.

Техническая поддержка

Для получения технической поддержки посетите раздел **Поддержка** сайта (www.kpda.ru). Вы обнаружите перечень предлагаемых способов оказания технической поддержки, включая публичный форум, форму обратной связи и контактные данные.

© 2017-2025, ООО «СВД ВС».

Обзор программного комплекса

Обзор программного комплекса поддержки работы с цифровой картографической информацией и общие рекомендации по применению на практике

- Работа с движком визуализации
- Параметры отображения
- Пользовательские объекты
- Преобразования и измерения
- Запуск и отображение

Программный комплекс позволяет создавать приложения, работающие с электронной картографической информацией (обработка, анализ, визуализация), под управлением ЗОСРВ «Нейтрино».

Разработка картографического приложения включает в себя следующие этапы:

1. подключение к картографическому сервису
2. подключение разделяемой области памяти (РОП)
3. настройка движка визуализации

Рассмотрим нюансы подключения к картографическому сервису. Для начала работы необходимо установить соединение с картографическим сервисом по идентификатору пользователя РОП.

```
#include <gis/gis.h>
#include <gis/gishelper.h>

gis_core_connection_t connection;
gis_core_link_init( &connection );

int result = gis_core_link_connect( &connection, 777 );
if ( result != EOK && result != EALREADY )
{
    printf( "Connection failed\n" );
    return result;
}
```

Существуют следующие возможности управления картографическим сервисом:

- запрос активных карт (во внутреннем формате):
 - фильтрация по границам;
 - фильтрация по масштабам карт;
- запрос списка классов по картам во внутреннем формате;
- управление кэшем:
 - полная перезапись всех карт внутреннего формата;
 - синхронизация карт во внутреннем формате с источником данных;
 - обновление метаданных карт внутреннего формата.

Data buffer - РОП, содержащая картографическую информацию во внутреннем формате. Для подключения РОП необходимо после предыдущего примера добавить вызов соответствующей функции:

```

...

result = gis_core_databuffer_attach( &connection );
if ( result != EOK && result != EALREADY )
{
    printf( "Databuffer attach failed\n" );
    return result;
}

```

После подключения РОП необходимо заполнить данными, для чего осуществляется запрос заполнения РОП. Функция `gis_core_databuffer_data_request()` позволяет запросить данные, удовлетворяющие ограничениям по координатам и масштабу.



Предельный размер РОП задается переменной окружения **GIS_CORE_DATABUFFER_SIZE_LIMIT**.

```

...

gis_core_request_parameters_t map; // Параметры запроса (границы, пределы масштабов)
gis_core_request_parameters_init( &map );

bool autosize = true; // Флаг автоматического расширения размера РОП
result = gis_core_databuffer_data_request( &connection, &map, autosize );
if ( result != EOK )
{
    if ( result != EOVERFLOW )
    {
        printf( "GIS Core map draw request failed\n" );
        return result;
    } else
        printf( "GIS Core map draw request EOVERFLOW\n" );
}

```

Движок визуализации в своей работе использует дескриптор РОП, для получения которого необходимо вызвать соответствующую функцию:

```

...

result = gis_core_databuffer_descriptor_attach( &connection );
if ( result != EOK && result != EALREADY )
{
    printf( "GIS Core draw buffer descriptor attach failed \n" );
    return result;
}

```

Перед началом работы с движком отображения необходимо проинициализировать его контекст:

```
...
```

```
gis_data_engine_context_t data_engine_ctx;  
gis_data_engine_alloc( map_widget_width, map_widget_height, GIS_DATA_ENGINE_MODE_SM,  
↳ GIS_DATA_ENGINE_BPP_32, &data_engine_ctx );
```

Параметры инициализации *map_widget_width*, *map_widget_height* задают размеры области отображения в пикселях (ширина, высота), аргумент **GIS_DATA_ENGINE_MODE_SM** устанавливает режим отображения Surface Manager, параметр **GIS_DATA_ENGINE_BPP_32** устанавливает режим цветности ARGB.

Работа с движком визуализации

Движок визуализации предоставляет пользователю необходимый для корректной работы с картографической информацией интерфейс. Существуют следующие возможности взаимодействия с движком визуализации:

- настройка параметров отображения;
- настройка слоев отображения;
- установка обработчиков callback-функций;
- выбор режима работы;
- рисование и масштабирование;
- настройка параметров отображения MTW;
- работа с пользовательскими объектами;
- выполнение преобразований координат и измерений;

Параметры отображения

Перед началом работы пользователь может настроить параметры отображения, указанные ниже:

```
...
```

```
gis_mdp_t disp_param = gis_data_engine_get_display_parameters( data_engine_ctx );  
  
/* Координаты центра области отображения (широта 40 градусов, долгота 45 градусов): */  
double_point_t center = { 45, 40 };  
gis_mdp_set_center_point( disp_param, center );  
gis_mdp_set_phys_scale( disp_param, real_scale_denom );  
  
/* Установка параметров проекции отображения (проекция Меркатора, эллипсоид WGS84): */  
gis_map_projection_t proj_params;  
gis_map_projection_init( &proj_params );  
proj_params.projection_idx = GIS_PROJECTION_MERC_PSEUDO;  
proj_params.ellipsoid_idx = GIS_ELLIPSOID_WGS84;  
gis_mdp_set_projection( disp_param, &proj_params );  
  
/* Установка разрешения устройства отображения: */  
gis_mdp_set_display_resolution( disp_param, DISPLAY_WIDTH_PX, DISPLAY_HEIGHT_PX );
```

```

/* Установка метрических размеров устройства отображения: */
gis_mdp_set_display_size( disp_param, DISPLAY_WIDTH_MM, DISPLAY_HEIGHT_MM);

/* Установка цвета фоновой заливки: */
gis_mdp_set_background_color( disp_param, 0xffffffff );

/* Установка яркости и контрастности рендеринга: */
gis_mdp_set_brightness_contrast( disp_param, br, cn );

```

После того, как был выбран конкретный движок визуализации (аргумент функции `gis_data_engine_alloc()`), необходимо проинициализировать контекст выбранного движка визуализации, а именно контекст движка «Surface Manager» (SM):

```

...

/* Контекст движка SM: */
gis_render_sm_context_t surface_manager_ctx;

/* Зона нечувствительности относительно размера поверхности: */
double blit_region_rate = 0.5;

/* Дополнение к размеру offscreen-поверхности: */
uint32_t offscreen_ext_px = 1000;

gis_render_sm_alloc( &surface_manager_ctx, data_engine_ctx, offscreen_ext_px,
↳ blit_region_rate, &connection );

```

Размер результирующей поверхности вычисляется по следующей формуле:

```

Size = (map_widget_width + offscreen_ext_px * 2) x (map_widget_height + offscreen_ext_px * 2);
Size = (640 + 1000 * 2) x (480 + 1000 * 2) = 2640 x 2480 px;

```

Коэффициент `blit_region_rate` отвечает за границы зоны нечувствительности. Границы этой зоны вычисляются следующим образом: размеры области отображения смещаются наружу на величину `blit_region_rate * offscreen_ext_px`. Как только область окна отображения выходит за границы этой зоны, вызывается перерисовка поверхности.



Рисунок 1. Зона нечувствительности и область буферизации.

Для отображения различных типов картографической информации существуют различные типы слоев движка SM:

GIS_RENDER_SM_LAYER_TYPE_DISABLED

отключение слоя;

GIS_RENDER_SM_LAYER_TYPE_MTW

слой растровых карт;

GIS_RENDER_SM_LAYER_TYPE_2DMAP

слой векторных карт;

GIS_RENDER_SM_LAYER_TYPE_USROBJ

слой пользовательских объектов.



Полный перечень типов слоев соответствует типу данных [gis_render_sm_layer_type_t](#).

Движок SM поддерживает различные режимы работы:

асинхронный

перерисовка без ожидания;

синхронный

ожидание пользователем завершения отрисовки;

полусинхронный

ожидание только на время отрисовки области виджета.

Для того, чтобы полноценно реализовать работу движка, необходимо переопределить обработчик события рисования виджета, а также создать обработчики событий изменения масштаба.

```
void MapWidget::paintEvent( QPaintEvent * event )
{
    /* Вызов перерисовки со сдвигами dx и dy: */
    int res;
    if ( (res = gis_render_sm_move( surface_manager_ctx, dx, dy )) != EOK )
    {
        if ( res == EAGAIN )
            printf( "Attempt to move during full render\n" );
    }

    dx = 0;
    dy = 0;

    /* Получение указателя на поверхность и приведение к типу QPixmap: */
    QPixmap pixmap = *(QPixmap *)gis_render_sm_get_view_pixmap( surface_manager_ctx );
    QPainter painter( this );
    painter.drawPixmap( 0, 0, pixmap );
}

void MapWidget::slotIncreaseScale()
{
    if ( g_new_scales )
    {
        ...

        /* Установка табличного значения знаменателя масштаба: */
        real_scale_denom = scale_values[idx_scale];
    } else {
        ...

        /* Установка специфического значения знаменателя масштаба: */
        real_scale_denom *= (1.0 - CUSTOM_SCALE_RATE);
    }

    /* Вызов перерисовки с новым установленным масштабом: */
    gis_render_sm_rescale( surface_manager_ctx, real_scale_denom );
}
```

В том случае, когда требуется работать с растровыми картами, необходимо включить соответствующий слой движка SM и настроить параметры отображения MTW. Пользователь может выбрать как дискретную заливку, так и градиентную, а также может предустановить диапазон высот, которые будут корректно отображены, установить цвет, которым будут отображаться высоты, выходящие за пределы, установленные пользователем.

Кроме того, если у пользователя нет необходимости ограничивать диапазон отображения, то он может установить режим автоматического изменения граничных высот.

```
/* Инициализация слоя растровых карт: */
gis_render_sm_set_layer_type( surface_manager_ctx, 0, GIS_RENDER_SM_LAYER_TYPE_MTW );

/* Установка цветовой палитры, предельных высот отображения, типа рисования: */
gis_mdp_set_raster_palette( disp_param, colorMap.data(), colorMap.size(),
↪ invalid_height_color, max_height, min_height, discretePaletteFlag );

/* Установка автоматического изменения граничных высот: */
gis_mdp_set_raster_height_mode( disp_param, autolimit_flag );
```

Пользовательские объекты

Комплекс предоставляет пользователю возможность работать с пользовательскими объектами, такими как полилиния, полигон, изображение. Есть возможность изменить цвет объекта и/или его границы, а также изменить вектор градусных точек, описывающих объект.

Типы	Добавление	Удаление	Изменение
Полилиния	+	+	+
Полигон	+	+	+
Изображение	+	+	+/-

Рисунок 2. Возможности работы с пользовательскими объектами.

```
/* Добавление объекта типа полигон: */
userobject obj;
obj = gis_render_sm_userdata_add_polygon( surface_manager_ctx, vector.data(), vector.size(),
↪ fill_color, border_color );

/* Изменение цвета заливки полигона и цвета его границы: */
gis_render_sm_userdata_edit_object_color( surface_manager_ctx, obj, new_color,
↪ new_border_color );

/* Вызов рисования поверхности, содержащей пользовательские объекты: */
gis_render_sm_redraw_userobject( surface_manager_ctx, true );
```

Преобразования и измерения

Комплекс позволяет преобразовать градусные координаты выбранной точки в метрические и обратно, экранные координаты в градусные и обратно. Кроме того, пользователь может измерять длину полилинии, описываемой градусными точками, на эллипсоиде, измерять площадь и периметр полигонов.



Полигон должен быть выпуклым.

```
/* Преобразование пиксельных координат в метрические: */
double_point_t meter_pnt;
int32_point_t px_pos = { 0, 0 };
gis_render_sm_convert_px2meters( surface_manager_ctx, px_pos, &meter_pnt );

/* Измерение длины линии, проложенной по градусным точкам: */
double distance = 0;
gis_render_sm_calculate_distance( surface_manager_ctx, degree_point_vector.data(),
    ↪ degree_point_vector.size(), &distance );
```

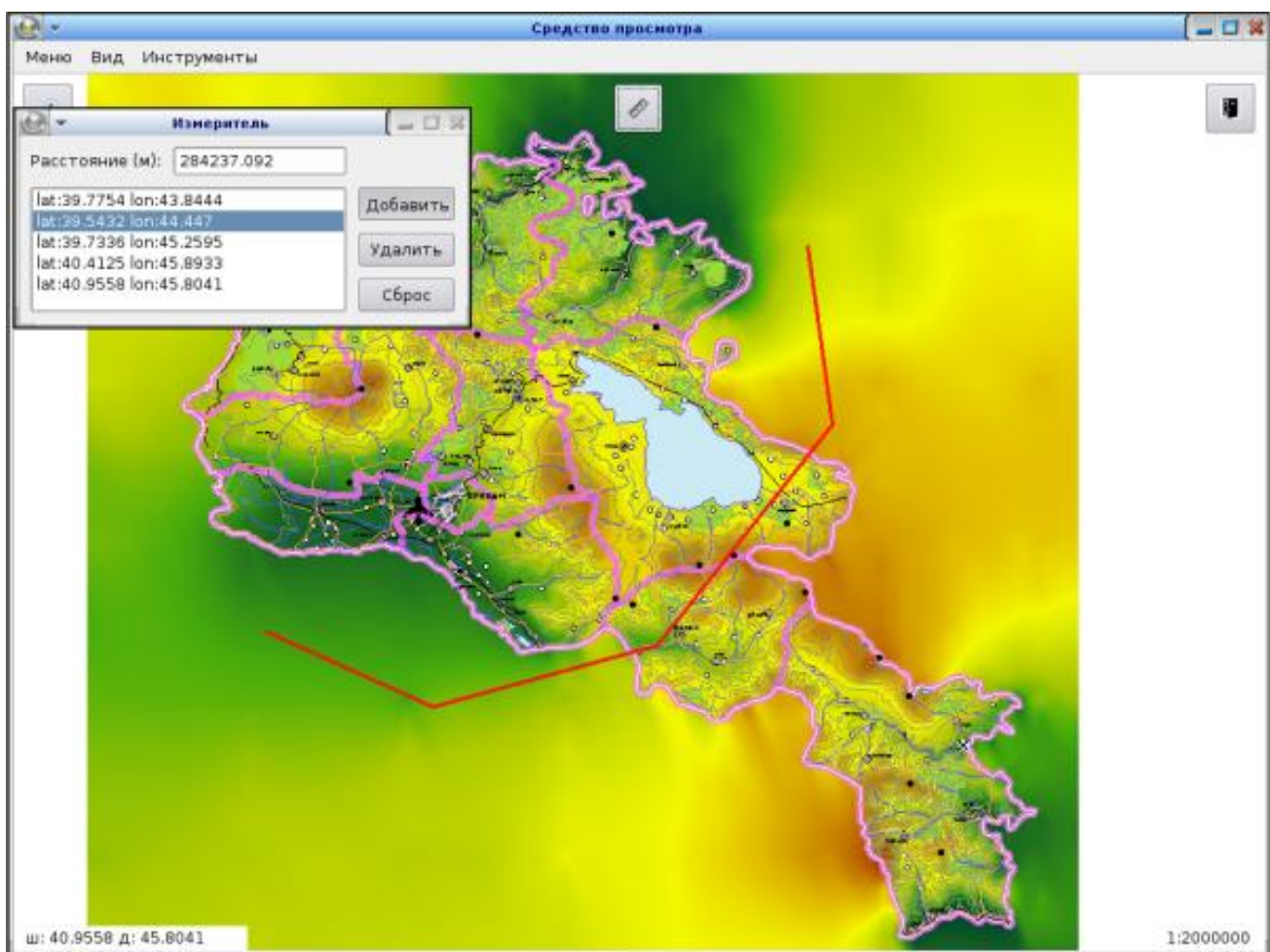


Рисунок 3. Измерение расстояний.

Запуск и отображение

Команда запуска картографического сервиса включает в себя название сервиса, а также список драйверов для определенных типов данных, которые необходимо подключить (работа с пользовательскими объектами включена по умолчанию).

```
/* Подключение драйверов SXF и MTW с локальным источником данных: */  
gis-core -d sxf-local, -d mtw-local
```

При запуске программного комплекса выполняется следующая последовательность действий:



Рисунок 4. Запуск картографического сервиса.

Команда запуска средства просмотра включает в себя название приложения, широту и долготу центра отображения.

```
gis-map-viewer -x45 -y40
```

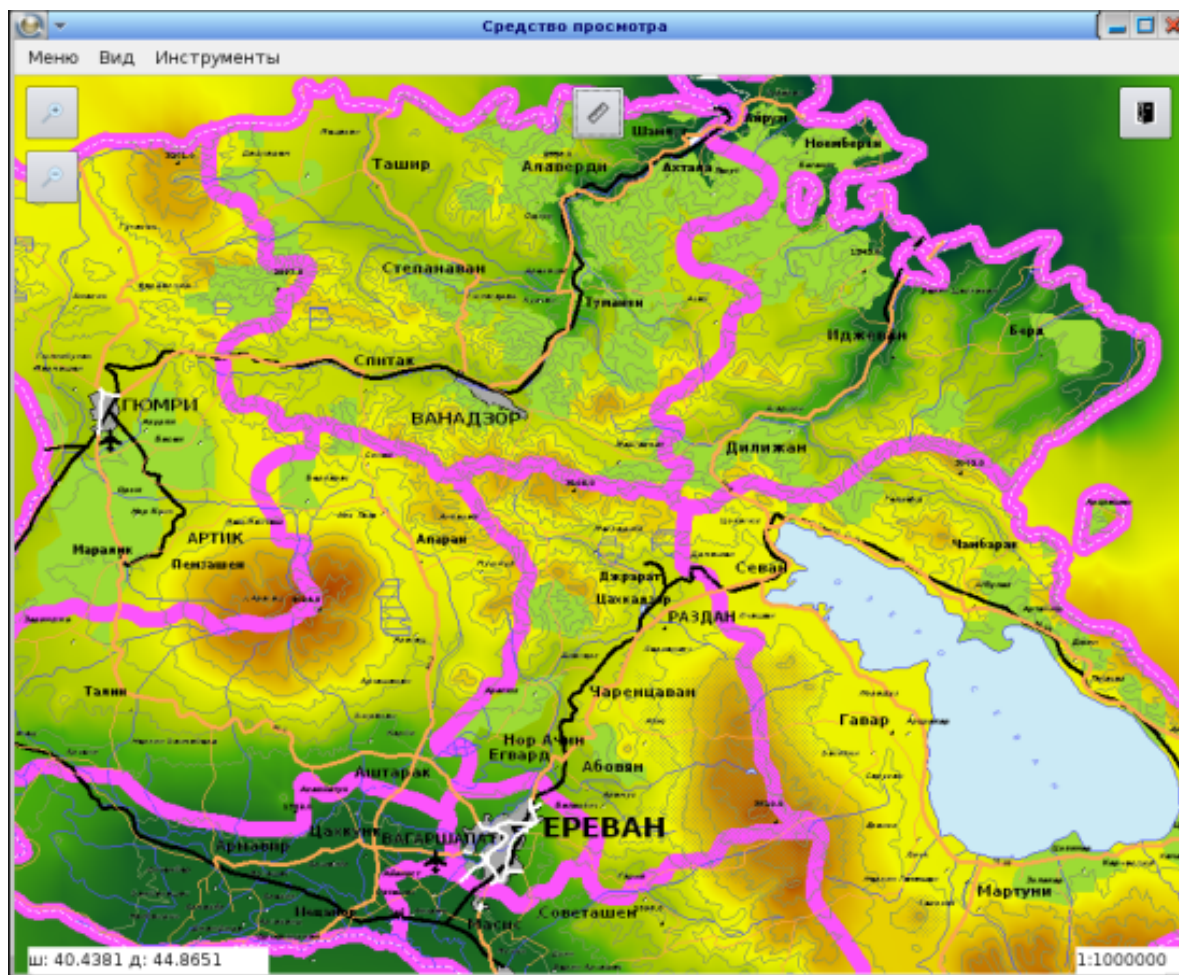


Рисунок 5. Средство просмотра.

Переменные окружения

Используемые переменные окружения и их краткое описание

- Общие переменные окружения
- Переменные окружения рендеринга

Общие переменные окружения

PATH

Путь поиска бинарных компонентов (переменная должна быть установлена в значение **\$GIS_ROOT/bin:\$GIS_ROOT/sbin:\$PATH**).

LD_LIBRARY_PATH

Путь поиска библиотек (переменная должна быть установлена в значение **\$GIS_ROOT/lib:\$LD_LIBRARY_PATH**).

GIS_ROOT

Корневая директория инсталляции компонентов (если переменная не задана используется путь /opt/gis).

GIS_DISABLE_VERIFIER

Отключение runtime-верификатора (если переменная задана со значением "y").

GIS_LOGGER_DIRECTORY

Корневая директория для хранения логов (если переменная не задана используется путь **\$GIS_ROOT/data/logs**).

GIS_CORE_MAP_CACHE

Корневая директория кэша карт (если переменная не задана используется путь **\$GIS_ROOT/data/maps/cache**).

GIS_CORE_DATABUFFER_SIZE_LIMIT

Предельный размер объекта в разделяемой памяти (по умолчанию используется 500 Мб).

GIS_DEBUG_LEVEL

Уровень подробности лога (0 - уровень ошибок (по-умолчанию), 1 - уровень предупреждений, 2 - уровень информационных сообщений, 3 - уровень отладочных сообщений).

GIS_DATABUFFER_LOGGING_ENABLE

Включение записи дескриптора разделяемой области памяти в файл (если переменная задана со значением "y").

GIS_RASTER_PROCESSOR_TMP_DIR

Директория для временного хранения перепроецированных растров в процессе работы утилиты [gis-raster-processor](#) с флагом **-c**.

Переменные окружения рендеринга

GIS_SM_GEN_THREADS

Предельное число потоков рендеринга для движка визуализации surface-malager (по умолчанию, а также при значении 0 будет установлено число потоков, равное числу доступных процессоров и/или процессорных ядер).

GIS_SM_GEN_TASKS_MULT

Множитель одновременно выполняемых задач потоком рендеринга для движка визуализации surface-manager (по умолчанию, а также при значении 0 будет установлено значение 3).

GIS_RENDERING_OPERATION_TIMEOUT_MS

Время ожидания завершения операции рендеринга в миллисекундах (по умолчанию значение равно 30000 мс).

GIS_SM_GEN_PRIORITY

Приоритет потоков рисования (по умолчанию значение равно 10).

Описание API

Описание API включает в себя группировку используемых функций и типов

Глава включает следующие разделы:

- Управление отладочным выводом
- Управление картографическим ядром (сервисом)
- Управление стилем отображения векторных карт
- Управление областью данных
- Управление движком визуализации (рендеринга)
 - Параметры отображения карты
 - Движок surfacemanager
 - Движок renderbuffer
- Общие функции и типы данных
- Дополнительные API
 - API для написания пользовательского драйвера форматов карт
 - Работа с конфигурационными файлами
 - Работа с таймером
 - Библиотека kd-tree

Управление отладочным выводом

Функция	Описание
<code>gis_helper_debug_file_getpath()</code>	Получение имени файла, в который направляется отладочный вывод.
<code>gis_helper_debug_file_setname()</code>	Задание имени файла, в который будет перенаправлен отладочный вывод.
<code>gis_helper_debug_mode_getmask()</code>	Получение маски отладочного вывода.
<code>gis_helper_debug_mode_name()</code>	Получение строки, описывающей текущий режим работы отладочного вывода.
<code>gis_helper_debug_mode_setlvl()</code>	Установка уровня подробности отладочного вывода.
<code>gis_helper_debug_mode_setmask()</code>	Установка режима работы отладочного вывода.
<code>gis_helper_debug_write()</code>	Вывод отладочного сообщения.

<code>gis_helper_debug_write_lvl()</code>	Вывод отладочного сообщения с заданным уровнем отладки.
<code>gis_helper_debug_write_mask()</code>	Вывод отладочного сообщения с заданной маской режима работы.

Тип	Описание
<code>gis_debug_level_t</code>	Уровни отладки.
<code>gis_debug_mode_t</code>	Маска режимов отладки.

Управление картографическим ядром (сервисом)

Функция	Описание
<code>gis_core_class_data_base_code()</code>	Декодирование кода класса, соответствующего оригиналу карты.
<code>gis_core_class_data_init()</code>	Инициализация информации о классе объекта карты.
<code>gis_core_class_list_free()</code>	Освобождение списка классов некоторой карты для выбранного драйвера.
<code>gis_core_class_list_init()</code>	Инициализация списка классов.
<code>gis_core_class_list_reinit()</code>	Повторная инициализация списка классов.
<code>gis_core_link_connect()</code>	Подключение к картографическому ядру.
<code>gis_core_link_destroy()</code>	Завершение открытой сессии с картографическим ядром.

<code>gis_core_link_init()</code>	Инициализация параметров дескриптора соединения с картографическим ядром.
<code>gis_core_map_list_free()</code>	Освобождение списка доступных для выбранного драйвера карт.
<code>gis_core_map_list_get_entry()</code>	Получение информации о доступной карте по ее идентификатору.
<code>gis_core_map_list_init()</code>	Инициализация списка карт.
<code>gis_core_request_connection_state()</code>	Запрос состояния соединения драйвера картографического ядра с источником данных.
<code>gis_core_request_download_maps()</code>	Запрос загрузки карт из источника данных.
<code>gis_core_request_driver_acronym()</code>	Запрос на получение акронима драйвера.
<code>gis_core_request_driver_connect()</code>	Запрос на установление соединения драйвера картографического ядра с источником данных (сервером).
<code>gis_core_request_driver_info()</code>	Запрос информации о конкретном драйвере картографического ядра.
<code>gis_core_request_drivers_prefix()</code>	Запрос префикса путей источников данных драйверов.
<code>gis_core_request_map_class_list()</code>	Получение списка классов некоторой карты для выбранного драйвера.
<code>gis_core_request_maps_list()</code>	Получение списка доступных карт для выбранного драйвера.

<code>gis_core_request_parameters_are_correct()</code>	Проверка параметров запроса картографической информации.
<code>gis_core_request_parameters_init()</code>	Инициализация параметров запроса к ядру.
<code>gis_core_request_parameters_set_borders()</code>	Установка границ региона запроса к картографическому ядру.
<code>gis_core_request_parameters_set_data_source()</code>	Установка типа запрашиваемых у картографического ядра данных.
<code>gis_core_request_parameters_set_driver()</code>	Установка типа драйвера в запросе к картографическому ядру.
<code>gis_core_request_revision()</code>	Получение номера версии и ревизии картографического ядра.
<code>gis_core_request_update_cache()</code>	Синхронизация изменений в картографическом кэше.
<code>gis_core_request_update_cached_map()</code>	Обновление карты в картографическом кэше.
<code>gis_helper_get_core_data_source_name()</code>	Получение строки с названием типа оригинала карты.
<code>gis_helper_get_map_source_by_driver_id()</code>	Получение типа оригинала карты по уникальному идентификатору драйвера источника данных.

<code>gis_helper_is_data_source_valid()</code>	Проверка типа оригинала карты.
--	--------------------------------

Тип	Описание
<code>gis_core_class_code_t</code>	Кода класс объектов в карте во внутреннем формате (GCM).
<code>gis_core_class_data_t</code>	Расширенное описание кода класса объектов в карте во внутреннем формате (GCM).
<code>gis_core_class_info_t</code>	Информация о классе объектов в карте во внутреннем формате (GCM).
<code>gis_core_class_list_t</code>	Список классов карты для выбранного драйвера
<code>gis_core_connection_state_t</code>	Состояние связи с картографическим ядром.
<code>gis_core_connection_t</code>	Дескриптор соединения с ядром картографического сервиса.
<code>gis_core_driver_id_t</code>	Уникальный идентификатор драйвера источника данных.
<code>gis_core_driver_info_t</code>	Структура информации о драйвере.
<code>gis_core_hash_t</code>	Хэш файла или карты.
<code>gis_core_map_data_source_t</code>	Тип оригинала карты во внутреннем формате (GCM).
<code>gis_core_map_information_t</code>	Информация о карте во внутреннем формате (GCM).

<code>gis_core_map_list_t</code>	Список карт, доступных для выбранного драйвера.
<code>gis_core_request_parameters_t</code>	Структура запроса к ядру картографического сервиса.
<code>gis_core_update_cache_mode_t</code>	Режимы обновления кэша данных.

Макрос	Описание
<code>GIS_CORE_DRIVER_FOR_LOOP_HEAD</code>	Обход всех доступных драйверов в цикле.
<code>GIS_CORE_DRIVER_ID_CHECK</code>	Контроль корректности идентификатора драйвера.
<code>GIS_IS_CORE_MAP_DATA_SOURCE_VECTOR</code>	Проверка, является ли выбранный источник данных векторным либо растровым.

Управление стилем отображения векторных карт

Функция	Описание
<code>gis_helper_get_map_style_by_data_source()</code>	Получение типа стиля карты по формату карты.
<code>gis_helper_get_map_style_name()</code>	Получение строки с названием типа стиля карты.
<code>gis_helper_is_map_style_valid()</code>	Проверка корректности типа стиля карты.

<code>gis_map_style_fill_active_classifier_name_list()</code>	Получение списка классификаторов отображаемых карт.
<code>gis_map_style_flip_language()</code>	Переключение языка карты на международный/национальный.
<code>gis_map_style_get_active_palette_idx()</code>	Получение индекса активной палитры.
<code>gis_map_style_get_palette_list()</code>	Получение списка палитр для выбранного стиля векторных карт.
<code>gis_map_style_get_SEABED()</code>	Получение параметров отображения глубины.
<code>gis_map_style_palette_get_entry_count()</code>	Получение размера списка палитр.
<code>gis_map_style_palette_get_entry_list()</code>	Получение непосредственно списка палитр из структуры.
<code>gis_map_style_palette_get_map_style()</code>	Получение стиля векторных карт из списка палитр.
<code>gis_map_style_palette_list_free()</code>	Освобождение списка палитр.
<code>gis_map_style_set_active_palette_idx()</code>	Установка индекса активной палитры.

<code>gis_map_style_set_SEABED()</code>	Установка параметров отображения глубины.
---	---

Тип	Описание
<code>gis_map_style_t</code>	Тип стиля карты.
<code>gis_vector_palette_list_t</code>	Список палитр стиля векторных карт.
<code>gis_vector_palette_t</code>	Дескриптор векторной палитры.

Управление областью данных

Функция	Описание
<code>gis_core_databuffer_attach()</code>	Функция подключения области данных к программе.
<code>gis_core_databuffer_check()</code>	Функция проверки области данных.
<code>gis_core_databuffer_data_request()</code>	Функция запроса данных у картографического ядра для выбранного региона.
<code>gis_core_databuffer_detach()</code>	Функция отключения области данных от программы.
<code>gis_core_databuffer_resize()</code>	Задание размера области данных.
<code>gis_core_databuffer_size()</code>	Функция запроса размера области данных.
<code>gis_object_free()</code>	Освобождение памяти объекта из области данных.

<code>gis_object_init()</code>	Инициализация объекта из области данных.
--------------------------------	--

Тип	Описание
<code>gis_databuffer_desc_t</code>	Дескриптор разделяемой области данных.

Класс/Метод	Описание
<code>class ClassifierInfo</code>	Класс, описывающий информацию о конкретном классификаторе.
<code>class ClassifierLayerInfo</code>	Класс, описывающий информацию о конкретном слое в классификаторе.
<code>class DBCallbacks</code>	Оповещение о состоянии разделяемой области памяти.
<code>class LayerInfo</code>	Класс, описывающий информацию о слоях и поверхностях Surface Manager.
<code>class MapInfo</code>	Класс, описывающий информацию о карте в РОП.
<code>class SMCallbacks</code>	Класс, предоставляющий интерфейс функций-обработчиков для движка рендеринга Surface Manager.
<code>DBCallbacks::databufferValidityChangedCallback()</code>	Функция оповещения о корректности РОП.
<code>SMCallbacks::errorCallback()</code>	Обработчик ошибок рисования.

<code>SMCallbacks::mapClassRequiredCallback()</code>	Флаг необходимости рисования класса объектов векторной карты.
<code>SMCallbacks::mapLayerCompleteCallback()</code>	Завершение рисования слоя объектов векторной карты.
<code>SMCallbacks::mapLayerRequiredCallback()</code>	Флаг необходимости рисования слоя объектов векторной карты.
<code>SMCallbacks::surfaceCompleteCallback()</code>	Завершение рисования поверхности.

Тип	Описание
<code>gis_object_t</code>	Информация об объекте из области данных.

Класс/Метод	Описание
<code>class GisObjectList</code>	Список географических объектов.
<code>GisObjectList::find_nearest_object()</code>	Функция поиска ближайшего к заданной точке объекта.
<code>GisObjectList::find_nearest_point()</code>	Функция поиска ближайшей точки объекта к заданной.
<code>GisObjectList::get_object()</code>	Функция, возвращающая объект в списке по индексу.

<code>GisObjectList::get_object_count()</code>	Функция, возвращающая количество объектов в списке.
<code>GisObjectList::~GisObjectList()</code>	Деструктор класса <code>GisObjectList</code> .
<code>GisObjectList::GisObjectList()</code>	Конструктор класса <code>GisObjectList</code> .

Управление движком визуализации (рендеринга)

Функция	Описание
<code>gis_data_engine_alloc()</code>	Создание контекста движка рендеринга.
<code>gis_data_engine_free()</code>	Освобождение контекста движка рендеринга.
<code>gis_data_engine_get_canvas_size()</code>	Получение размера окна отображения движка рендеринга.
<code>gis_data_engine_get_classifier_list()</code>	Получение списка классификаторов с информацией о слоях.
<code>gis_data_engine_get_class_list()</code>	Получение списка активных классов.
<code>gis_data_engine_get_display_parameters()</code>	Получение указателя на контекст параметров визуализации.
<code>gis_data_engine_get_map_list()</code>	Получение списка карт с привязкой к классификаторам.

<code>gis_data_engine_get_maps_projection()</code>	Получение информации о проекции отображения.
<code>gis_data_engine_set_canvas_size()</code>	Обновление размера окна отображения движка рендеринга.
<code>gis_data_engine_set_notify_func()</code>	Установка функции-обработчика на событие изменения состояния области данных.
<code>gis_data_engine_update()</code>	Обновление области данных движка рендеринга.
<code>gis_data_engine_validity_changed()</code>	Оповещение об изменении состояния движка рендеринга.

Тип	Описание
<code>gis_data_engine_context_t</code>	Контекст движка рендеринга.

Параметры отображения карты

Функция	Описание
<code>gis_gui_get_projection_parameters()</code>	Создание диалога установки проекции отображения.
<code>gis_gui_get_screen_parameters()</code>	Создание диалога установки параметров экрана.
<code>gis_map_ellipsoid_get_full_name()</code>	Получение названия эллипсоида по его индексу.

<code>gis_map_height_system_get_full_name()</code>	Получение названия системы высот по её индексу.
<code>gis_map_projection_get_full_name()</code>	Получение названия проекции по её индексу.
<code>gis_map_projection_has_zone()</code>	Проверка наличия у проекции деления на зоны.
<code>gis_map_projection_init()</code>	Инициализация параметров проекции карты.
<code>gis_map_projection_is_filled()</code>	Проверка расширенного контекста проекции карты.
<code>gis_map_projection_zero()</code>	Обнуление параметров проекции.
<code>gis_mdp_get_background_color()</code>	Получение цвета фона карты.
<code>gis_mdp_get_brightness_contrast()</code>	Получение значений яркости и контрастности отображения.
<code>gis_mdp_get_center_point_deg()</code>	Получение градусных координат центра отображения карты.
<code>gis_mdp_get_display_resolution()</code>	Получение установленного разрешения экрана в пикселях.
<code>gis_mdp_get_display_size()</code>	Получение установленного размера экрана в миллиметрах.
<code>gis_mdp_get_map_antialiasing_level()</code>	Получение текущего уровня сглаживания карты.

<code>gis_mdp_get_phys_scale()</code>	Получение текущего масштаба отображения карты.
<code>gis_mdp_get_projection()</code>	Получение текущей проекции отображения карты.
<code>gis_mdp_get_raster_color_mode()</code>	Получение типа заливки для рисования раstra.
<code>gis_mdp_get_raster_height_mode()</code>	Получение режима выбора границ высот для растровых файлов.
<code>gis_mdp_get_raster_invalid_height_color()</code>	Получение цвета отображения высот вне заданного диапазона.
<code>gis_mdp_get_raster_lower_height_limit()</code>	Получение нижнего предела высот отображения для растровых файлов.
<code>gis_mdp_get_raster_palette()</code>	Получение палитры для рисования раstra.
<code>gis_mdp_get_raster_upper_height_limit()</code>	Получение верхнего предела высот отображения для растровых файлов.
<code>gis_mdp_get_scaling_mode()</code>	Получение режима масштабирования для текущего отображения карты.
<code>gis_mdp_set_background_color()</code>	Установка цвета фона карты.
<code>gis_mdp_set_brightness_contrast()</code>	Установка яркости и контрастности отображения.
<code>gis_mdp_set_center_point()</code>	Установка центра отображения карты.

<code>gis_mdp_set_center_point_ptr()</code>	Установка указателя на центр отображения карты.
<code>gis_mdp_set_display_resolution()</code>	Установка разрешения экрана для текущего отображения карты.
<code>gis_mdp_set_display_size()</code>	Установка геометрических размеров экрана для текущего отображения карты.
<code>gis_mdp_set_map_antialiasing_level()</code>	Установка уровня сглаживания карты.
<code>gis_mdp_set_phys_scale()</code>	Установка масштаба текущего отображения карты.
<code>gis_mdp_set_projection()</code>	Установка проекции текущего отображения карты.
<code>gis_mdp_set_raster_color_mode()</code>	Установка типа заливки для рисования раstra.
<code>gis_mdp_set_raster_height_limits()</code>	Установка диапазона высот отображения для растровых файлов.
<code>gis_mdp_set_raster_height_mode()</code>	Установка режима выбора границ высот для растровых файлов.
<code>gis_mdp_set_raster_invalid_height_color()</code>	Установка цвета отображения высот вне заданного диапазона.
<code>gis_mdp_set_raster_palette()</code>	Установка палитры для рисования раstra.
<code>gis_mdp_set_scaling_mode()</code>	Установка режима масштабирования для текущего отображения карты.

Тип	Описание
<code>gis_map_ellipsoid_idx_t</code>	Индекс эллипсоида картографической информации.
<code>gis_map_height_system_idx_t</code>	Индекс системы высот картографической информации.
<code>gis_map_projection_idx_t</code>	Индекс проекции картографической информации.
<code>gis_map_projection_t</code>	Расширенный контекст проекции карты.
<code>gis_mdp_t</code>	Контекст параметров визуализации.

Движок surfacemanager

Функция	Описание
<code>gis_render_sm_alloc()</code>	Создание контекста движка рендеринга.
<code>gis_render_sm_calculate_distance()</code>	Вычисление длины траектории, описываемой градусными точками.
<code>gis_render_sm_calculate_polygon()</code>	Вычисление параметров полигона.
<code>gis_render_sm_convert_degree2fpx()</code>	Преобразование градусных координат точки в пиксельные координаты окна отображения с сохранением дробной части.
<code>gis_render_sm_convert_degree2px()</code>	Преобразование градусных координат точки в пиксельные координаты окна отображения.

<code>gis_render_sm_convert_fpx2degree()</code>	Преобразование пиксельных координат точки окна отображения в градусные с сохранением дробной части.
<code>gis_render_sm_convert_fpx2meters()</code>	Преобразование пиксельных координат точки окна отображения в метрические координаты с сохранением дробной части.
<code>gis_render_sm_convert_px2degree()</code>	Преобразование пиксельных координат точки окна отображения в градусные.
<code>gis_render_sm_convert_px2meters()</code>	Преобразование пиксельных координат точки окна отображения в метрические координаты.
<code>gis_render_sm_draw()</code>	Вызов обработчика рисования движка рендеринга Surface Manager.
<code>gis_render_sm_free()</code>	Освобождение контекста движка рендеринга Surface Manager.
<code>gis_render_sm_get_draw_canvas_borders()</code>	Получение значений границ холста в градусах.
<code>gis_render_sm_get_layer_type()</code>	Получение типа слоя, назначенного указанной поверхности.
<code>gis_render_sm_get_render_mode()</code>	Получение режима рисования движка рендеринга Surface Manager.
<code>gis_render_sm_get_view_pixmap()</code>	Получение поверхности окна отображения Surface Manager.
<code>gis_render_sm_move()</code>	Обработчик перемещения карты.
<code>gis_render_sm_redraw_userobject()</code>	Перерисовка пользовательского слоя.

<code>gis_render_sm_rescale()</code>	Масштабирование карты.
<code>gis_render_sm_set_layer_data_sources()</code>	Назначение форматов карт для отображения в заданной поверхности.
<code>gis_render_sm_set_layer_type()</code>	Назначение типа слоя заданной поверхности.
<code>gis_render_sm_set_render_mode()</code>	Установка режима рисования движка рендеринга Surface Manager.
<code>gis_render_sm_set_update_func()</code>	Установка функции оповещения о состоянии рисования Surface Manager.
<code>gis_render_sm_sync()</code>	Задать тип следующей операции рисования.
<code>gis_render_sm_update()</code>	Обновление параметров Surface Manager.
<code>gis_render_sm_userdata_add_bitmap()</code>	Добавление изображения.
<code>gis_render_sm_userdata_add_contour_object()</code>	Добавление точки контура.
<code>gis_render_sm_userdata_add_polygon()</code>	Добавление полигона.
<code>gis_render_sm_userdata_add_polyline()</code>	Добавление полилинии.

<code>gis_render_sm_userdata_delete_contour_object()</code>	Удаление точки контура.
<code>gis_render_sm_userdata_delete_object()</code>	Удаление пользовательского объекта.
<code>gis_render_sm_userdata_edit_object_color()</code>	Изменение цвета объекта.
<code>gis_render_sm_userdata_edit_object_points()</code>	Изменение точек объекта.

Тип	Описание
<code>gis_render_sm_context_t</code>	Контекст движка рендеринга Surface Manager.
<code>gis_render_sm_layer_type_t</code>	Слой движка рендеринга.
<code>gis_userobject_t</code>	Тип пользовательского объекта движка рендеринга Surface Manager.

Движок **renderbuffer**

Функция	Описание
<code>gis_render_rb_alloc()</code>	Создание контекста движка рендеринга.

<code>gis_render_rb_draw()</code>	Рисование карты.
<code>gis_render_rb_free()</code>	Освобождение контекста движка рендеринга.

Тип	Описание
<code>gis_render_rb_context_t</code>	Контекст движка рендеринга Render Buffer.

Общие функции и типы данных

Функция	Описание
<code>gis_data_raw_ctx_alloc()</code>	Создание контекста RAW для работы с областью данных.
<code>gis_data_raw_ctx_free()</code>	Удаление контекста RAW, необходимого для работы с областью данных.
<code>gis_data_raw_ctx_import_databuffer()</code>	Получение указателя на разделяемую область памяти.
<code>gis_data_raw_get_map_count()</code>	Получение количества карт в разделяемой области памяти.
<code>gis_data_raw_get_raster_block_count()</code>	Получение количества блоков растровых данных в текущей карте разделяемой области памяти
<code>gis_data_raw_get_raster_block_height()</code>	Получение высоты блока растровых данных

<code>gis_data_raw_get_raster_block_upper_left_corner()</code>	Получение координат верхнего левого угла указанного блока растровой карты (метры в проекции растра)
<code>gis_data_raw_get_raster_block_width()</code>	Получение ширины блока растровых данных
<code>gis_data_raw_get_raster_data()</code>	Получение данных блока растровых данных
<code>gis_data_raw_get_raster_height_in_blocks()</code>	Получение высоты текущей растровой карты в блоках
<code>gis_data_raw_get_raster_height_in_elements()</code>	Получение высоты растровой карты (в элементах)
<code>gis_data_raw_get_raster_meters_per_element()</code>	Получение разрешения растровой карты (метры на элемент)
<code>gis_data_raw_get_raster_upper_left_corner()</code>	Получение координат верхнего левого угла растровой карты (метры в проекции растра)
<code>gis_data_raw_get_raster_width_in_blocks()</code>	Получение ширины текущей растровой карты в блоках
<code>gis_data_raw_get_raster_width_in_elements()</code>	Получение ширины растровой карты (в элементах)

<code>gis_data_raw_map_get_borders()</code>	Получение градусных границ текущей карты в разделяемой области памяти
<code>gis_data_raw_map_get_data_source()</code>	Получение типа выбранной карты.
<code>gis_data_raw_map_get_filename()</code>	Получение названия текущей карты в разделяемой области памяти
<code>gis_data_raw_map_get_object_count()</code>	Получение количества объектов в текущей выбранной карте разделяемой области памяти.
<code>gis_data_raw_map_get_projection()</code>	Получение проекции текущей карты в разделяемой области памяти
<code>gis_data_raw_map_select_first_object()</code>	Выбор первого объекта в текущей карте разделяемой области памяти.
<code>gis_data_raw_map_select_next_object()</code>	Выбор следующего объекта в текущей карте разделяемой области памяти.
<code>gis_data_raw_map_select_object()</code>	Выбор объекта в текущей карте разделяемой области памяти по индексу.
<code>gis_data_raw_object_get_bounding_rect()</code>	Получение описывающего прямоугольника для объекта.
<code>gis_data_raw_object_get_class_code()</code>	Получение кода класса выбранного объекта текущей карты.

<code>gis_data_raw_object_get_data()</code>	Получение массива точек выбранного объекта текущей карты.
<code>gis_data_raw_object_get_height()</code>	Получение высоты выбранного объекта текущей карты.
<code>gis_data_raw_object_get_point_count()</code>	Получение количества точек в выбранном объекте текущей карты.
<code>gis_data_raw_object_get_type()</code>	Получение типа примитива выбранного объекта текущей карты.
<code>gis_data_raw_raster_maps_get_height_limits()</code>	Получение пределов высот для всех растровых карт в разделяемой области памяти.
<code>gis_data_raw_select_first_map()</code>	Выбор первой карты в разделяемой области памяти.
<code>gis_data_raw_select_map()</code>	Выбор требуемой карты в разделяемой области памяти.
<code>gis_data_raw_select_map_by_index()</code>	Выбор карты в разделяемой области памяти по индексу.
<code>gis_data_raw_select_next_map()</code>	Выбор следующей карты в разделяемой области памяти.
<code>gis_gui_about()</code>	Вывод информации о ПК ЦКИ.
<code>gis_helper_are_maps_equal()</code>	Сравнение двух карт.

<code>gis_helper_convert_point_degrees_2_meters()</code>	Преобразование координат точки (градусы) в координаты точки в метрической системе проекции.
<code>gis_helper_convert_point_meters_2_degrees()</code>	Преобразование координат точки в метрах (проекции) в координаты градусной меры.
<code>gis_helper_convert_point_meters_2_degrees_array()</code>	Преобразование координат массива точек в метрах (проекции) в координаты градусной меры.
<code>gis_helper_env_get_config_directory()</code>	Получение пути к директории, содержащей конфигурационные файлы.
<code>gis_helper_env_get_gis_root_directory()</code>	Получение текущего значения переменной окружения GIS_ROOT.
<code>gis_helper_env_get_maps_cache_directory()</code>	Получение текущего значения переменной окружения GIS_CORE↵_MAP_CACHE.
<code>gis_helper_free_string_list()</code>	Освобождение памяти списка строк.
<code>gis_helper_math_free_ctx()</code>	Освобождение математического контекста.
<code>gis_helper_math_generate_meters_projection()</code>	Заполнение метрических параметров математического контекста по переданным параметрам проекции.

<code>gis_helper_math_get_degrees_projection()</code>	Получение строки, описывающей градусную проекцию, в формате PROJ.4.
<code>gis_helper_math_get_meters_projection()</code>	Получение строки, описывающей метрическую проекцию, в формате PROJ.4.
<code>gis_helper_math_init_ctx()</code>	Инициализация математического контекста.
<code>gis_helper_split_string_by_delimiter()</code>	Разбить строку по разделителю.
<code>gis_object_primitive_type_get_full_name()</code>	Получение строки с названием типа примитива объекта карты.

Тип	Описание
<code>gis_borders_t</code>	Границы картографической информации.
<code>gis_data_raw_context_t</code>	Контекст низкоуровневого доступа к данным.
<code>gis_helper_math_ctx_t</code>	Контекст математических операций.
<code>gis_object_primitive_type_t</code>	Типы примитивов объекта карты.
<code>int32_point_t, float_point_t, double_point_t</code>	Представление примитива 'точка'.

<code>object_point_t</code>	Представление 'точки' объекта.
-----------------------------	--------------------------------

Макрос	Описание
<code>CLIP_VALUE</code>	Контроль нахождения в диапазоне между двумя числами.
<code>DEGREES_TO_RADS</code>	Конвертация градусных углов в радианы.
<code>EQUAL_POINTS_XY</code>	Проверка двух координат на равенство.
<code>EQUAL_POINTS_XY_PTRS</code>	Проверка двух координат на равенство с использованием указателей.
<code>GETSIGN</code>	Получение знака числа.
<code>RADS_TO_DEGREES</code>	Конвертация радиан в градусные углы.

Дополнительные API

Дополнительные API, предоставляющие расширенный функционал по использованию и анализу картографической информации.

API для написания пользовательского драйвера форматов карт

Функция	Описание
<code>gis_common_driver_check_folders</code>	Проверить наличие папок с картами и стилями.
<code>gis_common_driver_fill_extensions()</code>	Заполнить список расширений карт.

<code>gis_common_driver_get_basename_without_extension</code>	Получить базовое имя карты без расширения.
<code>gis_common_driver_get_mandatory_driver_param_from_cfg()</code>	Получить из файла конфигурации ядра один из основных (обязательных) параметров драйвера.
<code>gis_common_driver_recover_gcm_database()</code>	Восстановить базу данных GCM.
<code>gis_common_driver_reset_gcm_database()</code>	Очистить базу данных GCM.
<code>gis_common_driver_sync_cache()</code>	Синхронизировать кэш карт.
<code>gis_common_driver_update_cache()</code>	Переконвертировать карту в GCM.
<code>gis_common_driver_update_clipped_map_info()</code>	Обновить информацию о картах GCM с одноимённой исходной картой.

Тип	Описание
<code>gis_common_driver_extension_list_t</code>	Список возможных расширений файлов карт.
<code>gis_driver_ctx_t</code>	Контекст драйвера формата карт.

<code>gis_driver_interface_t</code>	Интерфейс драйвера формата исходной карты.
-------------------------------------	--

Макрос	Описание
<code>GIS_CACHE_SUBDIR_GCM</code>	Маркер названия директории, хранящей карты формата GCM.
<code>GIS_GCM_FILE_EXTENSION</code>	Маркер расширения карт формата GCM

Работа с конфигурационными файлами

Функция	Описание
<code>gis_core_config_get_driver_value()</code>	Получить из файла конфигурации ядра параметр драйвера.
<code>gis_core_config_get_value()</code>	Получить параметр из файла конфигурации ядра.
<code>gis_core_config_set_driver_value()</code>	Обновить значение параметра драйвера в файле конфигурации ядра.

Макрос	Описание
<code>DRIVER_CFG_*, DRIVER_*_CFG_SECTION</code>	Маркеры названий разделов драйверов в конфигурационном файле ядра.

Работа с таймером

Функция	Описание
---------	----------

<code>gis_helper_timer_get_result_sec()</code>	Получение значения таймера.
<code>gis_helper_timer_start()</code>	Запустить таймер.
<code>gis_helper_timer_stop()</code>	Остановить таймер.

Тип	Описание
<code>gis_timer_t</code>	Таймер.

Библиотека kd-tree

Функция	Описание
<code>gis_kd_add()</code>	Добавление нового узла.
<code>gis_kd_destroy()</code>	Удаление КД дерева.
<code>gis_kd_dnn()</code>	Поиск узлов в заданном радиусе.
<code>gis_kd_init()</code>	Создание КД дерева.
<code>gis_kd_knn()</code>	Поиск К ближайших узлов.
<code>gis_kd_rnn()</code>	Поиск узлов в заданном диапазоне.

Тип	Описание
<code>gis_kd_node_t</code>	Структура узла дерева.
<code>gis_kd_tree_t</code>	Структура КД дерева.
<code>gis_kd_uniq_id</code>	Структура идентификатора узла.

gis_helper_debug_file_getpath()

Получение имени файла, в который направляется отладочный вывод.

Прототип:

```
#include <gis/gishelper.h>

const char * gis_helper_debug_file_getpath();
```

Библиотека:

gishelper

Описание:

Функция возвращает имя файла, в который направляется отладочный вывод.

Примеры использования:

```
gis_helper_debug_mode_setmask( GIS_DEBUG_MODE_STD | GIS_DEBUG_MODE_FD );
gis_helper_debug_file_setname( "debug_file.log" );
const char *filepath = gis_helper_debug_file_getpath();

char manual_path[256];
sprintf( manual_path, "%s/data/logs/debug_file.log",
↪ gis_helper_env_get_gis_root_directory() );

if ( strcmp( manual_path, filepath ) != 0 )
{
    printf("Log files mismatch:\n%s\n%s\n", manual_path, filepath);
    return 1;
}
```

Классификация:

ПК ЦКИ для ЗОСРВ «Нейтрино»

Безопасность использования	
Обработчик прерываний	Нет
Обработчик сигналов	Нет
В потоке	Да

Тематические ссылки:

`gis_helper_debug_file_setname()`

gis_helper_debug_file_setname()

Задание имени файла, в который будет перенаправлен отладочный вывод.

Прототип:

```
#include <gis/gishelper.h>

int gis_helper_debug_file_setname( const char *name );
```

Аргументы:

name

Имя файла, в который будет перенаправлен отладочный вывод.

Библиотека:

gishelper

Описание:

Функция задаёт имя файла, в который будет перенаправлен отладочный вывод.

Возвращаемое значение:

EOK

Успешное завершение.

EINVAL

Не задано имя файла.

EIO

Не установлен флаг режима работы GIS_DEBUG_MODE_FD.

Примеры использования:

```
gis_helper_debug_mode_setmask( GIS_DEBUG_MODE_STD | GIS_DEBUG_MODE_FD );
gis_helper_debug_file_setname( "debug_file.log" );
const char *filepath = gis_helper_debug_file_getpath();

char manual_path[256];
sprintf( manual_path, "%s/data/logs/debug_file.log",
↪ gis_helper_env_get_gis_root_directory() );

if ( strcmp( manual_path, filepath ) != 0 )
```

```

{
    printf("Log files mismatch:\n%s\n%s\n", manual_path, filepath);
    return 1;
}

```

Классификация:

ПК ЦКИ для ЗОСРВ «Нейтрино»

Безопасность использования	
<i>Обработчик прерываний</i>	Нет
<i>Обработчик сигналов</i>	Нет
<i>В потоке</i>	Да

Тематические ссылки:

[gis_helper_debug_mode_getmask\(\)](#), [gis_helper_debug_file_getpath\(\)](#)

gis_helper_debug_mode_getmask()

Получение маски отладочного вывода.

Прототип:

```
#include <gis/gishelper.h>

gis_debug_mode_t gis_helper_debug_mode_getmask();
```

Библиотека:

gishelper

Описание:

Функция возвращает маску, описывающую текущий режим отладочного вывода (см GIS_DEBUG_MODE_*).

Примеры использования:

```
gis_debug_mode_t mode;
gis_helper_debug_mode_setmask( GIS_DEBUG_MODE_STD | GIS_DEBUG_MODE_FD );

mode = gis_helper_debug_mode_getmask();
if ( mode != ( GIS_DEBUG_MODE_STD | GIS_DEBUG_MODE_FD ) )
{
    printf( "Incorrect debug mode\n" );
    return 1;
}
```

Классификация:

ПК ЦКИ для ЗОСРВ «Нейтрино»

Безопасность использования	
Обработчик прерываний	Да
Обработчик сигналов	Да
В потоке	Да

Тематические ссылки:

gis_helper_debug_mode_setmask(), gis_helper_debug_mode_name(),

gis_helper_debug_mode_name()

Получение строки, описывающей текущий режим работы отладочного вывода.

Прототип:

```
#include <gis/gishelper.h>

const char * gis_helper_debug_mode_name( gis_debug_mode_t mask );
```

Аргументы:

mask

Маска режима типа `gis_debug_mode_t` (см. `GIS_DEBUG_MODE_*`).

Библиотека:

gishelper

Описание:

Функция возвращает строку, описывающую текущий режим работы отладочного вывода.

Возвращаемое значение:

!<none>

Успешное завершение.

<none>

Некорректный параметр функции.

Примеры использования:

```
gis_helper_debug_mode_setmask( GIS_DEBUG_MODE_STD );
const char *mode_name = gis_helper_debug_mode_name( gis_helper_debug_mode_getmask() );

if ( strcmp( mode_name, "std" ) != 0 )
{
    printf( "Incorrect debug mode\n" );
    return 1;
}
```

Классификация:

Безопасность использования	
<i>Обработчик прерываний</i>	Нет
<i>Обработчик сигналов</i>	Нет
<i>В потоке</i>	Да

Тематические ссылки:

[gis_helper_debug_mode_getmask\(\)](#), [gis_helper_debug_mode_setmask\(\)](#)

gis_helper_debug_mode_setlvl()

Установка уровня подробности отладочного вывода.

Прототип:

```
#include <gis/gishelper.h>

int gis_helper_debug_mode_setlvl( gis_debug_level_t lvl );
```

Аргументы:

mask

Уровень отладки типа `gis_debug_level_t` (см. `GIS_DEBUG_LEVEL_*`).

Библиотека:

gishelper

Описание:

Функция задаёт уровень подробности отладочного вывода.

Возвращаемое значение:

EOK

Успешное завершение.

EINVAL

Некорректные параметры функции.

Примеры использования:

```
gis_helper_debug_mode_setmask( GIS_DEBUG_MODE_STD | GIS_DEBUG_MODE_FD );
gis_helper_debug_mode_setlvl( GIS_DEBUG_LEVEL_INFO );

gis_helper_debug_write_lvl( GIS_DEBUG_LEVEL_INFO, "[GIS]: ", "Testing INFO setlvl" );
gis_helper_debug_write_lvl( GIS_DEBUG_LEVEL_DEBUG, "[GIS]: ", "Testing DEBUG setlvl" );

string line;
bool found_info = false;
bool found_debug = false;
ifstream logfile( gis_helper_debug_file_getpath() );
if ( logfile )
```

```

{
    while ( getline( logfile, line ) )
    {
        char *cstr = new char[line.length() + 1];
        strcpy( cstr, line.c_str() );

        char *istr = strstr( cstr, "Testing INFO setlvl" );
        if ( istr != NULL )
            found_info = true;

        istr = strstr( cstr, "Testing DEBUG setlvl" );
        if ( istr != NULL )
            found_debug = true;

        delete [] cstr;
    }

    if ( found_info == false || found_debug == true )
    {
        printf( "Wrong entries in log file\n" );
        return 1;
    }

    logfile.close();
}

```

Классификация:

ПК ЦКИ для ЗОСРВ «Нейтрино»

Безопасность использования	
Обработчик прерываний	Нет
Обработчик сигналов	Нет
В потоке	Да

Тематические ссылки:

[gis_helper_debug_mode_getmask\(\)](#), [gis_helper_debug_mode_name\(\)](#)

gis_helper_debug_mode_setmask()

Установка режима работы отладочного вывода.

Прототип:

```
#include <gis/gishelper.h>

int gis_helper_debug_mode_setmask( gis_debug_mode_t mask );
```

Аргументы:

mask

Маска режима типа `gis_debug_mode_t` (см. `GIS_DEBUG_MODE_*`).

Библиотека:

gishelper

Описание:

Функция задаёт режим работы отладочного вывода.

Возвращаемое значение:

EOK

Успешное завершение.

EINVAL

Некорректные параметры функции.

Примеры использования:

```
gis_debug_mode_t mode;
gis_helper_debug_mode_setmask( GIS_DEBUG_MODE_STD | GIS_DEBUG_MODE_FD );

mode = gis_helper_debug_mode_getmask();
if ( mode != ( GIS_DEBUG_MODE_STD | GIS_DEBUG_MODE_FD ) )
{
    printf( "Incorrect debug mode\n" );
    return 1;
}
```

Классификация:

ПК ЦКИ для ЗОСРВ «Нейтрино»

Безопасность использования	
Обработчик прерываний	Нет
Обработчик сигналов	Нет
В потоке	Да

Тематические ссылки:

`gis_helper_debug_mode_getmask()`, `gis_helper_debug_mode_name()`

gis_helper_debug_write()

Вывод отладочного сообщения.

Прототип:

```
#include <gis/gishelper.h>

int gis_helper_debug_write( const char *prefix, const char *format, ... );
```

Аргументы:

prefix

Префикс сообщения.

format

Описание форматирования строки.

...

Аргументы пользователя.

Библиотека:

gishelper

Описание:

Функция выводит отладочное сообщение в установленном ранее режиме работы и уровне отладки GIS_DEBUG_LEVEL_DEBUG.

Возвращаемое значение:

EOK

Успешное завершение.

ENOENT

Не удалось вывести сообщение в режиме GIS_DEBUG_MODE_STD.

ENODATA

Переданный уровень отладочного вывода меньше глобально заданного.

EFAULT

Не удалось открыть отладочный файл в режиме GIS_DEBUG_MODE_FD.

EBADFD

Не удалось вывести сообщение в файл отладочного вывода в режиме GIS_DEBUG_MODE_FD.

Примеры использования:

```
gis_helper_debug_mode_setmask( GIS_DEBUG_MODE_STD | GIS_DEBUG_MODE_FD );
gis_helper_debug_write( "[GIS]: ", ( char* )"Testing write" );

string line;
bool found = false;
ifstream logfile( gis_helper_debug_file_getpath() );
if ( logfile )
{
    while ( getline( logfile, line ) )
    {
        char *cstr = new char[line.length() + 1];
        strcpy( cstr, line.c_str() );

        char *istr = strstr( cstr, "Testing write" );
        if ( istr != NULL )
            found = true;

        delete [] cstr;
    }

    if ( found == false )
    {
        printf( "No such entry in log file\n" );
        return 1;
    }

    logfile.close();
}
```

Классификация:

ПК ЦКИ для ЗОСРВ «Нейтрино»

Безопасность использования	
Обработчик прерываний	Нет
Обработчик сигналов	Нет
В потоке	Да

Тематические ссылки:

[gis_helper_debug_write_mask\(\)](#), [gis_helper_debug_write_lvl\(\)](#)

gis_helper_debug_write_lvl()

Вывод отладочного сообщения с заданным уровнем отладки.

Прототип:

```
#include <gis/gishelper.h>

int gis_helper_debug_write_lvl( gis_debug_level_t lvl, const char *prefix, const char
↵ *format, ... );
```

Аргументы:

<i>lvl</i>	Уровень отладки.
<i>prefix</i>	Префикс сообщения.
<i>format</i>	Описание форматирования строки.
...	Аргументы пользователя.

Библиотека:

gishelper

Описание:

Функция выводит отладочное сообщение с заданным уровнем отладки в установленном ранее режиме.

Возвращаемое значение:

<i>EOK</i>	Успешное завершение.
<i>ENODATA</i>	Установленный ранее уровень отладки ниже требуемого.
<i>ENOENT</i>	Не удалось вывести сообщение в режиме GIS_DEBUG_MODE_STD.
<i>ESRCH</i>	Не удалось вывести сообщение в режиме GIS_DEBUG_MODE_SLOG.
<i>EFAULT</i>	Не удалось открыть отладочный файл в режиме GIS_DEBUG_MODE_FD.

Не удалось вывести сообщение в файл отладочного вывода в режиме GIS_DEBUG_MODE_FD.

Примеры использования:

```
gis_helper_debug_mode_setmask( GIS_DEBUG_MODE_STD | GIS_DEBUG_MODE_FD );
gis_helper_debug_mode_setlvl( GIS_DEBUG_LEVEL_INFO );

gis_helper_debug_write_lvl( GIS_DEBUG_LEVEL_INFO, "[GIS]: ", "Testing INFO write level" );
gis_helper_debug_write_lvl( GIS_DEBUG_LEVEL_DEBUG, "[GIS]: ", "Testing DEBUG write level"
↵ );

string line;
bool found_info = false;
bool found_debug = false;
ifstream logfile( gis_helper_debug_file_getpath() );
if ( logfile )
{
    while ( getline( logfile, line ) )
    {
        char *cstr = new char[line.length() + 1];
        strcpy( cstr, line.c_str() );

        char *istr = strstr( cstr, "Testing INFO write level" );
        if ( istr != NULL )
            found_info = true;

        istr = strstr( cstr, "Testing DEBUG write level" );
        if ( istr != NULL )
            found_debug = true;

        delete [] cstr;
    }

    if ( found_info == false || found_debug == true )
    {
        printf( "Wrong entries in log file\n" );
        return 1;
    }

    logfile.close();
}
```

Классификация:

ПК ЦКИ для ЗОСРВ «Нейтрино»

Безопасность использования	
<i>Обработчик прерываний</i>	Нет
<i>Обработчик сигналов</i>	Нет
<i>В потоке</i>	Да

Тематические ссылки:

`gis_helper_debug_write()`, `gis_helper_debug_write_mask()`

gis_helper_debug_write_mask()

Вывод отладочного сообщения с заданной маской режима работы.

Прототип:

```
#include <gis/gishelper.h>

int gis_helper_debug_write_mask( gis_debug_mode_t mask, const char *prefix, const char
↵ *format, ... );
```

Аргументы:

<i>mask</i>	Маска режима работы отладки.
<i>prefix</i>	Префикс сообщения.
<i>format</i>	Описание форматирования строки.
...	Аргументы пользователя.

Библиотека:

gishelper

Описание:

Функция выводит отладочное сообщение с заданной маской режима работы отладки и уровнем отладки GIS_DEBUG_LEVEL_DEBUG.

Возвращаемое значение:

<i>EOK</i>	Успешное завершение.
<i>ENOENT</i>	Не удалось вывести сообщение в режиме GIS_DEBUG_MODE_STD.
<i>ENODATA</i>	Установленный глобально уровень отладочного вывода меньше заданного.
<i>EFAULT</i>	Не удалось открыть отладочный файл в режиме GIS_DEBUG_MODE_FD.

EBADFD

Не удалось вывести сообщение в файл отладочного вывода в режиме GIS_DEBUG_MODE_FD.

Примеры использования:

```
gis_helper_debug_write_mask( GIS_DEBUG_MODE_STD | GIS_DEBUG_MODE_FD, "[GIS]: ", "Testing
↪ write mask" );

string line;
bool found = false;
ifstream logfile( gis_helper_debug_file_getpath() );
if ( logfile )
{
    while ( getline( logfile, line ) )
    {
        char *cstr = new char[line.length() + 1];
        strcpy( cstr, line.c_str() );

        char *istr = strstr( cstr, "Testing write mask" );
        if ( istr != NULL )
            found = true;

        delete [] cstr;
    }

    if ( found == false )
    {
        printf( "No such entry in log file\n" );
        return 1;
    }

    logfile.close();
}
```

Классификация:

ПК ЦКИ для ЗОСРВ «Нейтрино»

Безопасность использования	
Обработчик прерываний	Нет
Обработчик сигналов	Нет
В потоке	Да

Тематические ссылки:

[gis_helper_debug_write\(\)](#), [gis_helper_debug_write_lvl\(\)](#)

gis_debug_level_t

Уровни отладки.

Формат:

```
#include <gis/gishelper.h>

typedef enum {
    GIS_DEBUG_LEVEL_ERROR    = 0,
    GIS_DEBUG_LEVEL_WARNING,
    GIS_DEBUG_LEVEL_INFO,
    GIS_DEBUG_LEVEL_DEBUG,
    GIS_DEBUG_LEVEL_START    = GIS_DEBUG_LEVEL_ERROR,
    GIS_DEBUG_LEVEL_END      = GIS_DEBUG_LEVEL_DEBUG
} gis_debug_level_t;
```

Описание:

Перечисление gis_debug_level_t включает следующие значения:

- GIS_DEBUG_LEVEL_ERROR — вывод только критических сообщений об ошибках.
- GIS_DEBUG_LEVEL_WARNING — дополнение предыдущего режима предупреждениями.
- GIS_DEBUG_LEVEL_INFO — дополнение предыдущего режима информационными сообщениями.
- GIS_DEBUG_LEVEL_DEBUG — дополнение предыдущего режима отладочными сообщениями.

Текущий уровень отладки задаётся переменной окружения GIS_DEBUG_LEVEL ("0" соответствует GIS_DEBUG_LEVEL_ERROR, "3" - GIS_DEBUG_LEVEL_DEBUG). **Классификация:**

ПК ЦКИ для ЗОСРВ «Нейтрино»

Тематические ссылки:

gis_helper_debug_write_lvl()

gis_debug_mode_t

Маска режимов отладки.

Формат:

```
#include <gis/gishelper.h>

typedef uint32_t gis_debug_mode_t;
```

Описание:

Маска режимов отладки `gis_debug_mode_t` характеризует каналы вывода отладочной информации и является комбинацией следующих битов:

- `GIS_DEBUG_MODE_NONE` — вывод отладочной информации не осуществляется.
- `GIS_DEBUG_MODE_STD` — вывод отладочной информации в стандартные файлы вывода и ошибок.
- `GIS_DEBUG_MODE_SLOG` — вывод отладочной информации средствами системного логгера.
- `GIS_DEBUG_MODE_FD` — вывод отладочной информации в заданный файл.

Текущий уровень отладки задаётся переменной окружения `GIS_DEBUG_LEVEL` ("0" соответствует `GIS_DEBUG_LEVEL_ERROR`, "3" - `GIS_DEBUG_LEVEL_DEBUG`). **Классификация:**

ПК ЦКИ для ЗОСРВ «Нейтрино»

Тематические ссылки:

[gis_debug_level_t](#)

gis_core_class_data_base_code()

Декодирование кода класса, соответствующего оригиналу карты.

Прототип:

```
#include <gis/gis_types.h>

int gis_core_class_data_base_code( gis_core_class_data_t *data, uint64_t *base_code );
```

Аргументы:

data

Расширенный код класса, интерпретируемый в соответствии с конкретным источником данных.

base_code

Код класса, соответствующий оригиналу карты.

Библиотека:

gishelper

Описание:

Функция производит декодирование из расширенного исходного кода класса, соответствующего оригиналу карты. В рамках данного API подобные коды классов не используются.

Возвращаемое значение:

EOK

Успешное декодирование

EINVAL

Некорректно указан тип оригинала данных в расширенном коде класса

EFAULT

Переданы некорректные параметры

Примеры использования:

```

uint64_t code;
gis_core_class_data_t data;
gis_core_class_data_init( &data );
data.code = 0x2000003107450;
data.src = GIS_CORE_MAP_DATA_SOURCE_SXF;

gis_core_class_data_base_code( &data, &code );

long long converted_code = (long long)code;

if ( converted_code != 51410000 )
{
    printf( "Incorrect base code for an object. %lld\n", converted_code );
    return 1;
}

```

Классификация:

ПК ЦКИ для ЗОСРВ «Нейтрино»

Безопасность использования	
Обработчик прерываний	Нет
Обработчик сигналов	Нет
В потоке	Да

Поддерживается, начиная с ПК ЦКИ редакции 1.0.0

Тематические ссылки:

[gis_core_class_code_t](#), [gis_core_class_data_t](#)

gis_core_class_data_init()

Инициализация информации о классе объекта карты.

Прототип:

```
#include <gis/gis_types.h>

static inline void gis_core_class_data_init( gis_core_class_data_t *data);
```

Аргументы:

data

Расширенный код класса, интерпретируемый в соответствии с конкретным источником данных.

Библиотека:

gishelper

Описание:

Функция инициализирует информацию о классе объекта карты.

Примеры использования:

```
gis_core_class_data_t cdata;
gis_core_class_data_init( &cdata );

if ( cdata->code != GIS_CLASS_CODE_UNDEFINED ||
    cdata->src  != GIS_CORE_MAP_DATA_SOURCE_NONE )
{
    printf( "gis_core_class_data_init() failed\n" );
    return 1;
}
```

Классификация:

ПК ЦКИ для ЗОСРВ «Нейтрино»

Безопасность использования	
Обработчик прерываний	Нет
Обработчик сигналов	Нет
В потоке	Да

Тематические ссылки:

`gis_core_class_data_base_code()`

gis_core_class_list_free()

Освобождение списка классов некоторой карты для выбранного драйвера.

Прототип:

```
#include <gis/gishelper.h>

static inline void gis_core_class_list_free( gis_core_class_list_t *class_list );
```

Аргументы:

class_list

Список классов карты `gis_core_class_list_t`, сформированный функцией `gis_core_request_map_class_list()`.

Библиотека:

gishelper

Описание:

Функция освобождает память, задействованную для хранения списка доступных карт в формате `gis_core_class_list_t`, который был сформирован функцией `gis_core_request_map_class_list()`.

Примеры использования:

```
gis_core_class_list_t list;

gis_core_class_list_free( &list );
```

Классификация:

ПК ЦКИ для ЗОСРВ «Нейтрино»

Безопасность использования	
Обработчик прерываний	Нет
Обработчик сигналов	Да
В потоке	Да

Тематические ссылки:

`gis_core_class_list_t`, `gis_core_request_map_class_list()`

gis_core_class_list_init()

Инициализация списка классов.

Прототип:

```
#include <gis/gishelper.h>

static inline void gis_core_class_list_init( gis_core_class_list_t *list )
```

Аргументы:

list

Указатель на список классов.

Библиотека:

gishelper

Описание:

Функция позволяет инициализировать список классов.

Примеры использования:

```
gis_core_class_list_t list;
gis_core_class_list_init( &list );
```

Классификация:

ПК ЦКИ для ЗОСРВ «Нейтрино»

Безопасность использования	
Обработчик прерываний	Да
Обработчик сигналов	Да
В потоке	Да

Тематические ссылки:

[gis_core_class_list_t](#), [gis_core_class_list_init\(\)](#),

gis_core_class_list_reinit()

Повторная инициализация списка классов.

Прототип:

```
#include <gis/gishelper.h>

static inline void gis_core_class_list_reinit( gis_core_class_list_t *list )
```

Аргументы:

list

Указатель на список классов.

Библиотека:

gishelper

Описание:

Функция позволяет повторно инициализировать список классов.

Примеры использования:

```
gis_core_class_list_t list;
gis_core_class_list_reinit( &list );
```

Классификация:

ПК ЦКИ для ЗОСРВ «Нейтрино»

Безопасность использования	
Обработчик прерываний	Да
Обработчик сигналов	Да
В потоке	Да

Тематические ссылки:

[gis_core_class_list_t](#), [gis_core_class_list_init\(\)](#),

gis_core_link_connect()

Подключение к картографическому ядру.

Прототип:

```
#include <gis/gishelper.h>

int gis_core_link_connect( gis_core_connection_t *connection, int32_t shid );
```

Аргументы:

connection

Указатель на дескриптор соединения.

shid

Идентификатор разделяемой области памяти (РОП).

Библиотека:

gishelper

Описание:

Функция устанавливает соединение с картографическим ядром и возвращает информацию об открытой сессии через параметр *connection*. Для корректного завершения соединения должна использоваться функция [gis_core_link_destroy\(\)](#). При создании сессии картографическое ядро производит поиск разделяемого объекта по переданному идентификатору *shid*. Если объект с данным идентификатором ранее был создан, происходит его ассоциация с сессией клиента, в противном случае создается новый объект. В качестве *shid* может быть передано значение `SHID_NULL`. В этом случае картографическое ядро не будет ассоциировать с сессией ни один разделяемый объект и часть операций по работе с разделяемым объектом будут недоступны. Запросы к картографическому ядру, связанные с обслуживанием картографического кэша в данном случае преимущественно доступны.

Возвращаемое значение:

EOK

Успешное завершение.

EINVAL

Некорректный указатель на дескриптор соединения.

EPERM

Соединение запрещено.

ESRCH

Не удалось подключиться.

EALREADY

Повторное подключение.

EFAULT

Ошибка в ядре.

Примеры использования:

```
gis_core_connection_t    connection;
gis_core_link_init( &connection );
gis_core_link_connect( &connection, 777 );

if ( gis_core_request_connection_state( &connection, GIS_CORE_DRIVERS_SXF_LOCAL ) != EOK )
{
    printf( "Error: SXF driver connection state request failed\n" );
    return 1;
}

gis_core_link_destroy( &connection, true );
```

Классификация:

ПК ЦКИ для ЗОСРВ «Нейтрино»

Безопасность использования	
Обработчик прерываний	Нет
Обработчик сигналов	Нет
В потоке	Да

Тематические ссылки:

[gis_core_connection_t](#), [gis_core_link_init\(\)](#) [gis_core_link_destroy\(\)](#)

gis_core_link_destroy()

Завершение открытой сессии с картографическим ядром.

Прототип:

```
#include <gis/gishelper.h>

int gis_core_link_destroy( gis_core_connection_t *connection, bool free_shid );
```

Аргументы:

connection

Указатель на дескриптор соединения.

free_shid

Флаг запроса на освобождение ресурсов сессии, открытой по идентификатору.

Библиотека:

gishelper

Описание:

Функция завершает открытую картографическим ядром (сервисом) сессию. Установка флага `free_shid` указывает на удаление разделяемого объекта.

Возвращаемое значение:

EOK

Успешное завершение.

EINVAL

Некорректный указатель на дескриптор соединения.

ESRCH

Не удалось подключиться.

EFAULT

Дескриптор соединения не был проинициализирован.

Примеры использования:

```
gis_core_connection_t    connection;
gis_core_link_init( &connection );
gis_core_link_connect( &connection, 777 );

if ( gis_core_request_connection_state( &connection, GIS_CORE_DRIVERS_SXF_LOCAL ) != EOK )
{
    printf( "Error: SXF driver connection state request failed\n" );
    return 1;
}

gis_core_link_destroy( &connection, true );
```

Классификация:

ПК ЦКИ для ЗОСРВ «Нейтрино»

Безопасность использования	
Обработчик прерываний	Нет
Обработчик сигналов	Нет
В потоке	Да

Тематические ссылки:

[gis_core_connection_t](#), [gis_core_link_init\(\)](#) [gis_core_link_connect\(\)](#)

gis_core_link_init()

Инициализация параметров дескриптора соединения с картографическим ядром.

Прототип:

```
#include <gis/gishelper.h>

void gis_core_link_init( gis_core_connection_t *connection );
```

Аргументы:

req

Указатель на дескриптор соединения.

Библиотека:

gishelper

Описание:

Функция инициализирует параметры дескриптора соединения с картографическим ядром.

Примеры использования:

```
gis_core_connection_t  connection;
gis_core_link_init( &connection );
gis_core_link_connect( &connection, 777 );

if ( gis_core_request_connection_state( &connection, GIS_CORE_DRIVERS_SXF_LOCAL ) != EOK )
{
    printf( "Error: SXF driver connection state request failed\n" );
    return 1;
}

gis_core_link_destroy( &connection, true );
```

Классификация:

Безопасность использования	
Обработчик прерываний	Нет
Обработчик сигналов	Нет
В потоке	Да

Тематические ссылки:

[gis_core_connection_t gis_core_link_connect\(\)](#)

gis_core_map_list_free()

Освобождение списка доступных для выбранного драйвера карт.

Прототип:

```
#include <gis/gishelper.h>

void gis_core_map_list_free( gis_core_map_list_t *list )
```

Аргументы:

list

Список доступных карт [gis_core_map_list_t](#), сформированный функцией [gis_core_request_maps_list\(\)](#).

Библиотека:

gishelper

Описание:

Функция освобождает память, задействованную для хранения списка доступных карт в формате [gis_core_map_list_t](#), который был сформирован функцией [gis_core_request_maps_list\(\)](#).

Примеры использования:

```
gis_core_map_list_t list;
gis_core_map_list_init( &list );
gis_core_map_list_free( &list );
```

Классификация:

ПК ЦКИ для ЗОСРВ «Нейтрино»

Безопасность использования	
Обработчик прерываний	Нет
Обработчик сигналов	Да
В потоке	Да

Тематические ссылки:

[gis_core_map_information_t](#), [gis_core_map_list_t](#) [gis_core_request_maps_list\(\)](#)

gis_core_map_list_get_entry()

Получение информации о доступной карте по ее идентификатору.

Прототип:

```
#include <gis/gishelper.h>

gis_core_map_information_t *gis_core_map_list_get_entry( gis_core_map_list_t *map_list,
↪ uint32_t map_id );
```

Аргументы:

map_list

Список доступных карт `gis_core_map_list_t`, сформированный функцией `gis_core_request_maps_list()`.

map_id

Идентификатор карты.

Библиотека:

gishelper

Описание:

Функция производит поиск карты в списке доступных `map_list` и возвращает указатель на информационную структуру карты `gis_core_map_information_t` по уникальному идентификатору карты `map_id`.

Возвращаемое значение:

!NULL

Успешное завершение - возвращаемое значение соответствует указателю на информационную структуру карты с идентификатором `map_id`.

NULL

Возникла ошибка или `map_list` указывает на `NULL`.

Примеры использования:

```

gis_core_connection_t    connection;
gis_core_link_init( &connection );
gis_core_link_connect( &connection, 777 );

gis_core_request_parameters_t map_desc;
gis_core_request_parameters_init( &map_desc );

gis_core_map_list_t map_list;
gis_core_map_list_init( &map_list );

if ( gis_core_request_maps_list( &connection, &map_desc, GIS_CORE_DRIVERS_SXF_LOCAL,
↪ &map_list ) != EOK ) {
    printf( "gis_core_request_maps_list() failed\n" );
    return 1;
}

gis_core_map_information_t *map_information;
map_information = gis_core_map_list_get_entry( &map_list, 0 );

if ( map_information->id == 0 &&
    map_information->scale_denominator == 200000 &&
    map_information->object_count == 8589 )
{
    printf( "Correct\n" );
}
else
{
    printf( "Incorrect map parameters: %d %d %d", map_information->id,
↪ map_information->scale_denominator, map_information->object_count );
    return 1;
}

gis_core_map_list_free( &map_list );
gis_core_link_destroy( &connection, true );

```

Классификация:

ПК ЦКИ для ЗОСРВ «Нейтрино»

Безопасность использования	
Обработчик прерываний	Нет
Обработчик сигналов	Да
В потоке	Да

Тематические ссылки:

[gis_core_map_information_t](#), [gis_core_map_list_t](#) [gis_core_map_list_free\(\)](#)

gis_core_map_list_init()

Инициализация списка карт.

Прототип:

```
#include <gis/gishelper.h>

static inline void gis_core_map_list_init( gis_core_map_list_t *list )
```

Аргументы:

list

Указатель на список карт.

Библиотека:

gishelper

Описание:

Функция позволяет инициализировать список карт.

Примеры использования:

```
gis_core_map_list_t list;
gis_core_map_list_init( &list );
gis_core_map_list_free( &list );
```

Классификация:

ПК ЦКИ для ЗОСРВ «Нейтрино»

Безопасность использования	
Обработчик прерываний	Да
Обработчик сигналов	Да
В потоке	Да

Тематические ссылки:

[gis_core_map_list_t](#), [gis_core_map_list_free\(\)](#),

gis_core_request_connection_state()

Запрос состояния соединения драйвера картографического ядра с источником данных.

Прототип:

```
#include <gis/gishelper.h>

int gis_core_request_connection_state( gis_core_connection_t *connection,
↳ gis_core_driver_id_t driver_id );
```

Аргументы:

connection

Указатель на дескриптор соединения.

driver_id

Идентификатор драйвера.

Библиотека:

gishelper

Описание:

Функция позволяет определить состояние соединения с источником данных (сервером / хранилищем / базой данных / ...) для драйвера, определяемого идентификатором *driver_id* и дескриптором соединения с картографическим ядром (сервисом) *connection*.

Возвращаемое значение:

EOK

Соединение драйвера с источником данных находится в активном состоянии.

EINVAL

Некорректный указатель дескриптора соединения.

EAGAIN

Соединение драйвера с источником данных не установлено.

ESRCH

Запрос состояния соединения драйвера с источником данных не выполнен.

EPIPE

Соединение с картографическим ядром (сервисом) не установлено.

Примеры использования:

```
gis_core_connection_t connection;
gis_core_link_init( &connection );
gis_core_link_connect( &connection, 777 );

if ( gis_core_request_connection_state( &connection, GIS_CORE_DRIVERS_SXF_LOCAL ) != EOK )
{
    printf( "Error: driver connection state request failed\n" );
    return 1;
}

gis_core_link_destroy( &connection, true );
```

Классификация:

ПК ЦКИ для ЗОСРВ «Нейтрино»

Безопасность использования	
Обработчик прерываний	Нет
Обработчик сигналов	Нет
В потоке	Да

Тематические ссылки:

[gis_core_connection_t](#), [gis_core_driver_id_t](#) [gis_core_request_driver_connect\(\)](#)

gis_core_request_download_maps()

Запрос загрузки карт из источника данных.

Прототип:

```
#include <gis/gishelper.h>

int gis_core_request_download_maps( gis_core_connection_t *connection,
↵ gis_core_request_parameters_t *map );
```

Аргументы:

connection

Указатель на дескриптор соединения.

map

Параметры запроса.

Библиотека:

gishelper

Описание:

Функция отправляет запрос на загрузку карт из источника данных

Возвращаемое значение:

EOK

Карты загружены.

EINVAL

Некорректные аргументы

ENODATA

Отсутствуют требуемые карты.

ECONNREFUSED

Соединение отклонено.

ESRCH

Запрос не выполнен.

EPIPE

Соединение с картографическим ядром не установлено.

Примеры использования:

```
gis_core_connection_t connection;  
gis_core_link_init( &connection );  
gis_core_link_connect( &connection, 777 );  
  
gis_core_request_parameters_t map_desc;  
gis_core_request_parameters_init( &map_desc );  
gis_core_request_parameters_set_driver( &map_desc, GIS_CORE_DRIVERS_SXF_LOCAL );  
  
gis_core_request_download_maps( &connection, &map_desc );  
  
gis_core_link_destroy( &connection, true );
```

Классификация:

ПК ЦКИ для ЗОСРВ «Нейтрино»

Безопасность использования	
Обработчик прерываний	Нет
Обработчик сигналов	Нет
В потоке	Да

Тематические ссылки:

[gis_core_connection_t](#), [gis_core_request_parameters_t](#) [gis_core_request_parameters_set_borders\(\)](#)

gis_core_request_driver_acronym()

Запрос на получение акронима драйвера.

Прототип:

```
#include <gis/gishelper.h>

int gis_core_request_driver_acronym( gis_core_connection_t *connection,
↪ gis_core_driver_id_t driver_id, char* acronym );
```

Аргументы:

connection

Указатель на дескриптор соединения.

driver_id

Идентификатор драйвера.

acronym

Строка для хранения полученного акронима.

Библиотека:

gishelper

Описание:

Функция позволяет получить акроним для драйвера, определяемого идентификатором `driver_id` и дескриптором соединения с картографическим ядром (сервисом) `connection`.

Возвращаемое значение:

EOK

Успешное завершение.

EFAULT

Не удалось получить акроним.

EINVAL

Некорректные входные параметры.

Примеры использования:

```

int result;
gis_core_driver_id_t driver_id = GIS_CORE_DRIVERS_SXF_LOCAL;
char acronym[GIS_MAX_ACRONYM_LENGTH];
gis_core_connection_t connection;
gis_core_link_init( &connection );
gis_core_link_connect( &connection, 777 );

if ( ( result = gis_core_request_driver_acronym( &connection, driver_id, acronym ) ) != EOK
↪ )
{
    printf( "GIS Core request acronym failed - driver id = %d\n", driver_id );
    return 1;
}

if ( strcmp( acronym, "SXF local" ) != 0 )
{
    printf( "Incorrect acronym - %s", acronym );
    return 1;
}

gis_core_link_destroy( &connection, true );

```

Классификация:

ПК ЦКИ для ЗОСРВ «Нейтрино»

Безопасность использования	
Обработчик прерываний	Нет
Обработчик сигналов	Нет
В потоке	Да

Тематические ссылки:

[gis_core_connection_t](#), [gis_core_driver_id_t](#)

gis_core_request_driver_connect()

Запрос на установление соединения драйвера картографического ядра с источником данных (сервером).

Прототип:

```
#include <gis/gishelper.h>

int gis_core_request_driver_connect( gis_core_connection_t *connection,
↵ gis_core_driver_id_t driver_id );
```

Аргументы:

connection

Указатель на дескриптор соединения.

driver_id

Идентификатор драйвера.

Библиотека:

gishelper

Описание:

Функция позволяет установить соединение с источником данных (сервером / хранилищем / базой данных / ...) для драйвера, определяемого идентификатором *driver_id* и дескриптором соединения с картографическим ядром (сервисом) *connection*.

Возвращаемое значение:

EOK

Соединение драйвера с источником данных установлено.

EAGAIN

Соединение драйвера с источником данных не установлено.

ESRCH

Установление соединения драйвера с источником данных не выполнено.

EPIPE

Соединение с картографическим ядром не установлено.

Примеры использования:

```

gis_core_connection_t connection;
gis_core_link_init( &connection );
gis_core_link_connect( &connection, 777 );

if ( gis_core_request_driver_connect( &connection, GIS_CORE_DRIVERS_SXF_LOCAL ) != EOK ) {
    printf( "Status: connection not established\n" );
    return 1;
}

if ( gis_core_request_connection_state( &connection, GIS_CORE_DRIVERS_SXF_LOCAL ) != EOK )
{
    printf( "Error: driver connection state request failed\n" );
    return 1;
}

gis_core_link_destroy( &connection, true );

```

Классификация:

ПК ЦКИ для ЗОСРВ «Нейтрино»

Безопасность использования	
Обработчик прерываний	Нет
Обработчик сигналов	Да
В потоке	Да

Тематические ссылки:

[gis_core_connection_t](#), [gis_core_driver_id_t](#) [gis_core_request_connection_state\(\)](#)

gis_core_request_driver_info()

Запрос информации о конкретном драйвере картографического ядра.

Прототип:

```
#include <gis/gishelper.h>

int gis_core_request_driver_info( gis_core_connection_t *connection, gis_core_driver_info_t
↵ *driver_info );
```

Аргументы:

connection

Указатель на дескриптор соединения.

driver_info

Указатель на структуру для хранения информации о драйвере.

Библиотека:

gishelper

Описание:

Функция позволяет отправить запрос в картографическое ядро о предоставлении информации о драйвере. Для успешного выполнения запроса требуется предварительное заполнение поля `driver_id` структуры `driver_info`.

Возвращаемое значение:

EOK

Соединение драйвера с источником данных установлено.

ESRCH

Запрос не выполнен.

EPIPE

Соединение с картографическим ядром не установлено.

EINVAL

Некорректные аргументы.

Примеры использования:

```

gis_core_connection_t    connection;
gis_core_driver_info_t  driver_info;
driver_info.driver_id = GIS_CORE_DRIVERS_SXF_LOCAL;
gis_core_link_init( &connection );
gis_core_link_connect( &connection, 777 );

if ( gis_core_request_driver_info( &connection, &driver_info ) != EOK ) {
    printf( "Request failed\n" );
    return 1;
}

if ( driver_info.driver_id != GIS_CORE_DRIVERS_SXF_LOCAL ||
    driver_info.initialized != true ) {
    printf( "Driver info is incorrect\n" );
    return 1;
}

gis_core_link_destroy( &connection, true );

```

Классификация:

ПК ЦКИ для ЗОСРВ «Нейтрино»

Безопасность использования	
Обработчик прерываний	Нет
Обработчик сигналов	Да
В потоке	Да

Тематические ссылки:

[gis_core_connection_t](#), [gis_core_driver_info_t](#), [gis_core_link_connect\(\)](#)

gis_core_request_drivers_prefix()

Запрос префикса путей источников данных драйверов.

Прототип:

```
#include <gis/gishelper.h>

int gis_core_request_drivers_prefix( gis_core_connection_t *connection, char *out_dr_prefix
↵ );
```

Аргументы:

connection

Указатель на дескриптор соединения.

out_dr_prefix

Указатель на буфер, в который будет записан путь до папок драйвера (рекомендуется иметь величину буфера не менее GIS_MAX_PATH_LENGTH).

Библиотека:

gishelper

Описание:

Функция позволяет запросить префикс пути, по которому ядро производит поиск оригинальных файлов, распределенных по драйверам, а также генерацию файлов GCM. Установка префикса, по которым ядро производит поиск папок драйверов, возможна с использованием переменной окружения GIS_CORE_MAP_CACHE. Значение по умолчанию: /opt/gis/data/maps/cache.

Возвращаемое значение:

EOK

Соединение драйвера с источником данных установлено.

ESRCH

Запрос не выполнен.

EPIPE

Соединение с картографическим ядром не установлено.

EINVAL

Некорректные аргументы.

Примеры использования:

```

gis_core_connection_t    connection;
char                    prefix[GIS_MAX_PATH_LENGTH];
gis_core_link_init( &connection );
gis_core_link_connect( &connection, 777 );

if ( gis_core_request_drivers_prefix( &connection, prefix ) != EOK ) {
    printf( "gis_core_request_drivers_prefix() failed\n" );
    return 1;
}

if ( strcmp( gis_helper_env_get_maps_cache_directory(), prefix ) != 0 ) {
    printf( "Incorrect result: %s %s\n", gis_helper_env_get_maps_cache_directory(), prefix
↵ );
    return 1;
}

gis_core_link_destroy( &connection, true );

```

Классификация:

ПК ЦКИ для ЗОСРВ «Нейтрино»

Безопасность использования	
Обработчик прерываний	Нет
Обработчик сигналов	Нет
В потоке	Да

Тематические ссылки:

[gis_core_connection_t](#), [gis_core_link_connect\(\)](#)

gis_core_request_map_class_list()

Получение списка классов некоторой карты для выбранного драйвера.

Прототип:

```
#include <gis/gishelper.h>

int gis_core_request_map_class_list( gis_core_connection_t *connection,
↪ gis_core_map_information_t *map, gis_core_class_list_t *class_list );
```

Аргументы:

connection

Указатель на дескриптор соединения.

map_id

Идентификатор карты.

class_list

Указатель на список классов карты `gis_core_class_list_t`, сформированный функцией `gis_core_request_map_`

Библиотека:

gishelper

Описание:

Функция позволяет запросить у картографического ядра список классов карты во с идентификатором `map_id`. При успешном выполнении запроса функция выделяет память для возвращаемого списка классов `class_list` самостоятельно.

Возвращаемое значение:

EOK

Успешное завершение.

EPIPE

Соединение с картографическим ядром (сервисом) не установлено.

EINVAL

Один из аргументов `connection` или `class_list` указывает на NULL.

ESRCH

Получение списка классов карты не выполнено.

EFAULT

Чтение списка классов карты завершилось с ошибкой.

Примеры использования:

```
gis_core_connection_t    connection;

gis_core_link_init( &connection );
gis_core_link_connect( &connection, 777 );

gis_core_request_parameters_t map_desc;
gis_core_request_parameters_init( &map_desc );

gis_core_databuffer_attach( &connection );
gis_core_databuffer_data_request( &connection, &map_desc, 1 );

gis_core_map_list_t map_list;
gis_core_map_list_init( &map_list );
gis_core_request_maps_list( &connection, &map_desc, GIS_CORE_DRIVERS_SXF_LOCAL, &map_list
↵ );

gis_core_class_list_t    list;
gis_core_class_list_init( &list );
if ( gis_core_request_map_class_list( &connection, &map_list.entry_list[0], &list ) != EOK
↵ ) {
    printf( "gis_core_request_map_class_list() failed\n" );
    return 1;
}

if ( list.entry_count != 169 )
{
    printf( "Incorrect number of classes: %d\n", list.entry_count );
    return 1;
}

gis_core_map_list_free( &map_list );
gis_core_link_destroy( &connection, true );
```

Классификация:

ПК ЦКИ для ЗОСРВ «Нейтрино»

Безопасность использования	
Обработчик прерываний	Нет
Обработчик сигналов	Да
В потоке	Да

Тематические ссылки:

[gis_core_connection_t](#), [gis_core_class_list_t](#), [gis_core_driver_id_t](#), [gis_core_class_list_free\(\)](#)

gis_core_request_maps_list()

Получение списка доступных карт для выбранного драйвера.

Прототип:

```
#include <gis/gishelper.h>

int gis_core_request_maps_list( gis_core_connection_t *connection,
↪ gis_core_request_parameters_t *params, gis_core_driver_id_t driver_id, gis_core_map_list_t
↪ *map_list );
```

Аргументы:

connection

Указатель на дескриптор соединения.

params

Указатель на структуру параметров запроса к картографическому ядру.

driver_id

Идентификатор драйвера.

map_list

Указатель на список карт.

Библиотека:

gishelper

Описание:

Функция позволяет запросить у картографического ядра список карт во внутреннем формате (GCM), которые доступны драйверу *driver_id* для заданного региона *params* (анализируется лишь поле *borders* данной структуры). При успешном выполнении запроса функция выделяет память для возвращаемого списка карт *map_list* самостоятельно.

Возвращаемое значение:

EOK

Успешное завершение.

ESRCH

Запрос не выполнен.

EPIPE

Соединение с картографическим ядром не установлено.

EINVAL

Передан некорректный указатель.

EFAULT

Чтение списка карт завершилось с ошибкой.

ENOMEM

Не удалось выделить память для списка карт.

ENODATA

Отсутствуют данные, удовлетворяющие параметрам запроса.

Примеры использования:

```
gis_core_connection_t    connection;
gis_core_link_init( &connection );
gis_core_link_connect( &connection, 777 );

gis_core_request_parameters_t map_desc;
gis_core_request_parameters_init( &map_desc );

gis_core_map_list_t map_list;
gis_core_map_list_init( &map_list );

if ( gis_core_request_maps_list( &connection, &map_desc, GIS_CORE_DRIVERS_SXF_LOCAL,
↪ &map_list ) != EOK ) {
    printf( "gis_core_request_maps_list() failed\n" );
    return 1;
}

if ( map_list.entry_count != 2 ) {
    printf( "Incorrect number of maps: %d\n", map_list.entry_count );
    return 1;
}

gis_core_map_list_free( &map_list );
gis_core_link_destroy( &connection, true );
```

Классификация:

ПК ЦКИ для ЗОСРВ «Нейтрино»

Безопасность использования	
Обработчик прерываний	Нет
Обработчик сигналов	Да
В потоке	Да

Тематические ссылки:

gis_core_connection_t, gis_core_request_parameters_t, gis_core_driver_id_t gis_core_map_list_t gis_core_map_list_fre
gis_core_map_list_get_entry()

gis_core_request_parameters_are_correct()

Проверка параметров запроса картографической информации.

Прототип:

```
#include <gis/gishelper.h>

int gis_core_request_parameters_are_correct( gis_core_request_parameters_t *params );
```

Аргументы:

params

Указатель на структуру параметров запроса.

Библиотека:

gishelper

Описание:

Функция позволяет проверить параметры запроса картографической информации.

Возвращаемое значение:

0

Некорректные параметры.

!0

Корректные параметры запроса.

Примеры использования:

```
gis_core_request_parameters_t params;
gis_core_request_parameters_init( &params );

if ( gis_core_request_parameters_are_correct( &params ) == 0 )
{
    printf( "Error initializing parameters\n" );
    return 1;
}
```

Классификация:

Безопасность использования	
Обработчик прерываний	Нет
Обработчик сигналов	Нет
В потоке	Да

Поддерживается, начиная с ПК ЦКИ редакции 1.1.0

Тематические ссылки:

`gis_core_request_parameters_t`, `gis_borders_t`, `gis_core_request_parameters_set_data_source()`

gis_core_request_parameters_init()

Инициализация параметров запроса к ядру.

Прототип:

```
#include <gis/gishelper.h>

static inline void gis_core_request_parameters_init( gis_core_request_parameters_t *req )
```

Аргументы:

req

Указатель на структуру параметров запроса.

Библиотека:

gishelper

Описание:

Функция инициализирует структуру параметров запроса карт.

Примеры использования:

```
gis_core_request_parameters_t req;
gis_core_request_parameters_init( &req );
```

Классификация:

ПК ЦКИ для ЗОСРВ «Нейтрино»

Безопасность использования	
Обработчик прерываний	Нет
Обработчик сигналов	Нет
В потоке	Да

Тематические ссылки:

[gis_core_request_parameters_t](#)

gis_core_request_parameters_set_borders()

Установка границ региона запроса к картографическому ядру.

Прототип:

```
#include <gis/gishelper.h>

int gis_core_request_parameters_set_borders( gis_core_request_parameters_t *req,
↪ gis_borders_t *borders );
```

Аргументы:

req

Указатель на структуру параметров запроса.

borders

Указатель на границы области.

Библиотека:

gishelper

Описание:

Функция позволяет изменить границы региона, по которому запрашивается картографическая информация.

Возвращаемое значение:

EOK

Успешное завершение.

EINVAL

Некорректные параметры запроса.

Примеры использования:

```

gis_core_request_parameters_t req;
gis_core_request_parameters_init( &req );
gis_borders_t req_borders = { degrees: { west: 20,
                                          south: 30,
                                          east: 190,
                                          north: 40 } } };

gis_core_request_parameters_set_borders( &req, &req_borders );

if ( fabs( req.borders.degrees.west - 20.000000 ) < 0.00001 &&
    fabs( req.borders.degrees.south - 30.000000 ) < 0.00001 &&
    fabs( req.borders.degrees.east - 180.000000 ) < 0.00001 &&
    fabs( req.borders.degrees.north - 40.000000 ) < 0.00001 )
{
    printf( "Correct\n" );
}
else
{
    printf( "Incorrect borders: %lf %lf %lf %lf\n", req.borders.degrees.west,
                                                  req.borders.degrees.south,
                                                  req.borders.degrees.east,
                                                  req.borders.degrees.north );

    return 1;
}

```

Классификация:

ПК ЦКИ для ЗОСРВ «Нейтрино»

Безопасность использования	
Обработчик прерываний	Нет
Обработчик сигналов	Нет
В потоке	Да

Тематические ссылки:

[gis_core_request_parameters_t](#), [gis_borders_t](#), [gis_core_request_parameters_set_data_source\(\)](#)

gis_core_request_parameters_set_data_source()

Установка типа запрашиваемых у картографического ядра данных.

Прототип:

```
#include <gis/gishelper.h>

static inline int gis_core_request_parameters_set_data_source(
↪ gis_core_request_parameters_t *req, gis_core_map_data_source_t data_source )
```

Аргументы:

req

Указатель на структуру параметров запроса (`gis_core_request_parameters_t`).

data_source

Тип картографических данных (`gis_core_map_data_source_t`).

Библиотека:

gishelper

Описание:

Функция позволяет изменять тип запрашиваемой картографической информации.

Возвращаемое значение:

EOK

Успешное завершение.

EINVAL

Некорректные параметры запроса.

Примеры использования:

```
gis_core_request_parameters_t map;
gis_core_request_parameters_init( &map );
gis_core_request_parameters_set_data_source( &map, GIS_CORE_MAP_DATA_SOURCE_SXF );
```

Классификация:

Безопасность использования	
Обработчик прерываний	Нет
Обработчик сигналов	Нет
В потоке	Да

Тематические ссылки:

`gis_core_request_parameters_t`, `gis_core_map_data_source_t`, `gis_core_request_parameters_set_driver()`

gis_core_request_parameters_set_driver()

Установка типа драйвера в запросе к картографическому ядру.

Прототип:

```
#include <gis/gishelper.h>

static inline int gis_core_request_parameters_set_driver( gis_core_request_parameters_t
↪ *req, gis_core_driver_id_t dr_id )
```

Аргументы:

req

Указатель на структуру параметров запроса (`gis_core_request_parameters_t`).

data_source

Тип картографических данных (`gis_core_map_data_source_t`).

Библиотека:

gishelper

Описание:

Функция позволяет изменять тип драйвера, по которому запрашивается картографическая информация.

Возвращаемое значение:

EOK

Успешное завершение.

EINVAL

Некорректные параметры запроса.

Примеры использования:

```
gis_core_request_parameters_t req;
gis_core_request_parameters_init( &req );
gis_core_request_parameters_set_driver( &req, GIS_CORE_DRIVERS_SHP_LOCAL );
```

Классификация:

Безопасность использования	
Обработчик прерываний	Нет
Обработчик сигналов	Нет
В потоке	Да

Тематические ссылки:

`gis_core_request_parameters_t`, `gis_core_driver_id_t`, `gis_core_request_parameters_set_data_source()`

gis_core_request_revision()

Получение номера версии и ревизии картографического ядра.

Прототип:

```
#include <gis/gishelper.h>

int gis_core_request_revision( gis_core_connection_t *connection, int *revision, char
↵ *version );
```

Аргументы:

connection

Указатель на дескриптор соединения.

revision

Указатель на номер ревизии.

version

Указатель на номер версии.

Библиотека:

gishelper

Описание:

Функция позволяет запросить у картографического ядра номер версии и ревизии.

Возвращаемое значение:

EOK

Успешное завершение.

EINVAL

Один из аргументов указывает на NULL.

EFAULT

Запрос ревизии картографического ядра завершился с ошибкой.

Примеры использования:

```
gis_core_connection_t    connection;

gis_core_link_init( &connection );
gis_core_link_connect( &connection, 777 );

int revision;
char version[GIS_MAX_NAME_LENGTH];
if ( gis_core_request_revision( &connection, &revision, &version[0] ) != EOK )
{
    printf( "Failed to get revision\n" );
    return 1;
}

gis_core_link_destroy( &connection, true );
```

Поддерживается, начиная с ПК ЦКИ редакции 1.2

Классификация:

ПК ЦКИ для ЗОСРВ «Нейтрино»

Безопасность использования	
Обработчик прерываний	Нет
Обработчик сигналов	Да
В потоке	Да

Тематические ссылки:

[gis_core_connection_t](#),

gis_core_request_update_cache()

Синхронизация изменений в картографическом кэше.

Прототип:

```
#include <gis/gishelper.h>

int gis_core_request_update_cache( gis_core_connection_t *connection, gis_core_driver_id_t
↪ driver_id, gis_core_update_cache_mode_t mode );
```

Аргументы:

connection

Указатель на дескриптор соединения.

driver_id

Идентификатор драйвера.

mode

Режим обновления кэша карт.

Библиотека:

gishelper

Описание:

Функция позволяет отправить в картографическое ядро запрос на обновление или сброса содержимого картографического кэша. Способ модификации содержимого картографического кэша определяется режимом *mode*. Для повторного наполнения разделяемой памяти удаленными или новыми картами требуется запросить их у сервиса штатными средствами. Необходимо отдельно отметить, что полностью разделяемый объект этой функцией не очищается, в нем остаются карты, которые не были определены картографическим ядром (сервисом) как изменившиеся.

Возвращаемое значение:

EOK

Соединение драйвера с источником данных установлено.

ESRCH

Запрос не выполнен.

EPIPE

Соединение с картографическим ядром не установлено.

Примеры использования:

```
gis_core_connection_t connection;
gis_core_link_init( &connection );
gis_core_link_connect( &connection, 777 );

if ( gis_core_request_update_cache( &connection, GIS_CORE_DRIVERS_SXF_LOCAL,
↳ GIS_CORE_UPDATE_CACHE_MODE_HARD_RESET ) != EOK ) {
    printf( "Failed to update\n" );
    return 1;
}

gis_core_link_destroy( &connection, true );
```

Классификация:

ПК ЦКИ для ЗОСРВ «Нейтрино»

Безопасность использования	
Обработчик прерываний	Нет
Обработчик сигналов	Да
В потоке	Да

Тематические ссылки:

[gis_core_connection_t](#), [gis_core_driver_id_t](#) [gis_core_update_cache_mode_t](#) [gis_core_request_update_cached_map\(\)](#)

gis_core_request_update_cached_map()

Обновление карты в картографическом кэше.

Прототип:

```
#include <gis/gishelper.h>

int gis_core_request_update_cached_map( gis_core_connection_t *connection,
↪ gis_core_driver_id_t driver_id, int32_t map_id );
```

Аргументы:

connection

Указатель на дескриптор соединения.

driver_id

Идентификатор драйвера.

map_id

Идентификатор карты в контексте драйвера.

Библиотека:

gishelper

Описание:

Функция позволяет отправить в картографическое ядро (сервис) запрос на обновление карты в картографическом кэше. Идентификатор карты `map_id` драйвера `driver_id` может быть определен с помощью функции `gis_core_request_maps_list()`. Для более мягкой синхронизации изменений в локальном хранилище или для повторной перекодировки всей совокупности картографической информации рекомендуется использовать функцию `gis_core_request_update_cache()`.

Обновление карты включает:

- Удаление карты в разделяемом объекте (повторную запись данная функция не осуществляет, для этого ее потребуется запросить у сервиса штатными средствами).
- Удаление преобразованной во внутренний формат GCM копии данной карты.
- Удаление всех хранящихся ядром сведений о карте.
- Синхронизация изменений, связанных с данной картой.

Возвращаемое значение:

ЕОК

Соединение драйвера с источником данных установлено.

EINVAL

Некорректный аргумент.

ESRCH

Запрос не выполнен.

EPIPE

Соединение с картографическим ядром не установлено.

Примеры использования:

```
gis_core_connection_t connection;
gis_core_link_init( &connection );
gis_core_link_connect( &connection, 777 );

if ( gis_core_request_update_cached_map( &connection, GIS_CORE_DRIVERS_SXF_LOCAL, 0 ) !=
↪ EOK ) {
    printf( "Failed to update map\n" );
    return 1;
}

gis_core_link_destroy( &connection, true );
```

Классификация:

ПК ЦКИ для ЗОСРВ «Нейтрино»

Безопасность использования	
Обработчик прерываний	Нет
Обработчик сигналов	Да
В потоке	Да

Считается устаревшим, начиная с ПК ЦКИ редакции 1.2

Тематические ссылки:

[gis_core_connection_t](#), [gis_core_driver_id_t](#), [gis_core_request_maps_list\(\)](#), [gis_core_request_update_cache\(\)](#)

gis_helper_get_core_data_source_name()

Получение строки с названием типа оригинала карты.

Прототип:

```
#include <gis/gis_types.h>

static inline const char * gis_helper_get_core_data_source_name( gis_core_map_data_source_t
↵ data_source );
```

Аргументы:

data_source

Индекс типа данных карты.

Библиотека:

gishelper

Описание:

Функция возвращает статическую строку, описывающую название типа оригинала карты.

Возвращаемое значение:

!NULL

Строка с названием типа оригинала карты

NULL

Передан некорректный data_source

Примеры использования:

```
gis_core_map_data_source_t src = GIS_CORE_MAP_DATA_SOURCE_SXF;

if ( strcmp( gis_helper_get_core_data_source_name( src ), "SXF" ) != 0 )
{
    printf( "Data source name is invalid\n" );
    return 1;
}
```

Классификация:

ПК ЦКИ для ЗОСРВ «Нейтрино»

Безопасность использования	
<i>Обработчик прерываний</i>	Нет
<i>Обработчик сигналов</i>	Нет
<i>В потоке</i>	Да

Тематические ссылки:

[gis_core_map_data_source_t](#)

gis_helper_get_map_source_by_driver_id()

Получение типа оригинала карты по уникальному идентификатору драйвера источника данных.

Прототип:

```
#include <gis/gis_types.h>

static inline gis_core_map_data_source_t gis_helper_get_map_source_by_driver_id(
↪ gis_core_driver_id_t driver_id );
```

Аргументы:

driver_id

Идентификатор драйвера источника данных.

Библиотека:

gishelper

Описание:

Функция возвращает типа оригинала карты, соответствующий уникальному идентификатору драйвера источника данных.

Возвращаемое значение:

!GIS_CORE_MAP_DATA_SOURCE_END

Тип оригинала карты

GIS_CORE_MAP_DATA_SOURCE_END

Задан некорректный идентификатор драйвера

Примеры использования:

```
gis_core_driver_id_t driver_id = GIS_CORE_DRIVERS_S57_LOCAL;

if ( gis_helper_get_map_source_by_driver_id( driver_id ) != GIS_CORE_MAP_DATA_SOURCE_S57 )
{
    printf( "Map source is invalid\n" );
    return 1;
}
```

Классификация:

ПК ЦКИ для ЗОСРВ «Нейтрино»

Безопасность использования	
Обработчик прерываний	Да
Обработчик сигналов	Да
В потоке	Да

Тематические ссылки:

[gis_core_driver_id_t](#), [gis_core_map_data_source_t](#)

gis_helper_is_data_source_valid()

Проверка типа оригинала карты.

Прототип:

```
#include <gis/gis_types.h>

static inline bool gis_helper_is_data_source_valid( gis_core_map_data_source_t src );
```

Аргументы:

src

Индекс типа данных карты.

Библиотека:

gishelper

Описание:

Функция проверяет типа оригинала карты на корректность.

Возвращаемое значение:

true

Тип оригинала карты заполнен успешно

false

Тип оригинала карты не доступен для использования

Примеры использования:

```
gis_core_map_data_source_t src = GIS_CORE_MAP_DATA_SOURCE_SXF;

if ( !gis_helper_is_data_source_valid( src ) )
{
    printf( "Data source is invalid" );
    return 1;
}
```

Классификация:

ПК ЦКИ для ЗОСРВ «Нейтрино»

Безопасность использования	
Обработчик прерываний	Да
Обработчик сигналов	Да
В потоке	Да

Тематические ссылки:

[gis_core_map_data_source_t](#)

gis_core_class_code_t

Кода класс объектов в карте во внутреннем формате (GCM).

Формат:

```
#include <gis/gis_types.h>

typedef uint64_t    gis_core_class_code_t;
```

Описание:

Код класса, интерпретируемый в соответствии с терминологией формата оригинальной карты. В случае, если код класса у объекта не определен используется константа GIS_CLASS_CODE_UNDEFINED.

Классификация:

ПК ЦКИ для ЗОСРВ «Нейтрино»

Тематические ссылки:

[gis_core_class_data_t](#), [gis_core_class_list_t](#), [gis_core_class_info_t](#)

gis_core_class_data_t

Расширенное описание кода класса объектов в карте во внутреннем формате (GCM).

Формат:

```
#include <gis/gis_types.h>

typedef struct
{
    gis_core_class_code_t    code;
    gis_core_map_data_source_t  src;
} gis_core_class_data_t;
```

Описание:

Структура содержит описание расширенного кода класса объектов карты во внутреннем формате (GCM) и включает следующие поля:

Аргументы:

code

Код класса.

src

Тип оригинала карты во внутреннем формате.

Классификация:

ПК ЦКИ для ЗОСРВ «Нейтрино»

Тематические ссылки:

[gis_core_class_code_t](#), [gis_core_map_data_source_t](#)

gis_core_class_info_t

Информация о классе объектов в карте во внутреннем формате (GCM).

Формат:

```
#include <gis/gishelper.h>

typedef struct {
    gis_core_class_data_t    class_data;
    char                    acronym[GIS_MAX_ACRONYM_LENGTH];
    uint32_t                object_count;
} gis_core_class_info_t;
```

Описание:

Структура `gis_core_class_info_t` содержит характеристики класса объектов карты во внутреннем формате (GCM) и включает следующие поля:

Аргументы:

class_data

Код класса, интерпретируемый в соответствии как с особенностями формата оригинала карты, так и с оптимизациями драйвера картографического ядра (сервера).

acronym

Акроним класса (массив символов размером `GIS_MAX_ACRONYM_LENGTH`).

object_count

Количество объектов класса.

Классификация:

ПК ЦКИ для ЗОСРВ «Нейтрино»

Тематические ссылки:

[gis_core_class_list_t](#), [gis_core_class_data_t](#),

gis_core_class_list_t

Список классов карты для выбранного драйвера

Формат:

```
#include <gis/gishelper.h>

typedef struct {
    gis_core_class_info_t    *entry_list;
    uint32_t                 entry_count;
} gis_core_class_list_t;
```

Описание:

Структура `gis_core_class_list_t` представляет список классов некоторой карты, доступной для выбранного драйвера, и содержит следующие поля:

Аргументы:

entry_list

Динамический массив записей обо всех доступных в карте классах. Формат записи определяется типом `gis_core_class_info_t`.

entry_count

Количество классов

Классификация:

ПК ЦКИ для ЗОСРВ «Нейтрино»

Тематические ссылки:

`gis_core_class_info_t`, `gis_core_request_map_class_list()`, `gis_core_class_list_free()`,

gis_core_connection_state_t

Состояние связи с картографическим ядром.

Формат:

```
#include <gis/gis_types.h>

typedef enum {
    CONNECTED = 1,
    NO_CONNECT = 2
} gis_core_connection_state_t;
```

Описание:

Перечисление `gis_core_connection_state_t` описывает состояние связи с картографическим ядром. Перечисление включает следующие состояния:

- `CONNECTED` — клиент подключён к ядру
- `NO_CONNECT` — клиент не подключён к ядру

Классификация:

ПК ЦКИ для ЗОСРВ «Нейтрино»

Поддерживается, начиная с ПК ЦКИ редакции 1.1.0

gis_core_connection_t

Дескриптор соединения с ядром картографического сервиса.

Формат:

```
#include <gis/gishelper.h>

typedef struct gis_core_connection {
    const char    *name;
    int32_t       link;
    uint32_t       state;
    int32_t       shid;
} gis_core_connection_t;
```

Описание:

Структура `gis_core_connection_t` описывает дескриптор соединения с ядром, поля интерпретируются следующим образом:

Аргументы:

<i>name</i>	Имя соединения.
<i>link</i>	Идентификатор соединения.
<i>state</i>	Текущее состояние соединения.
<i>shid</i>	Идентификатор разделяемой области памяти (РОП).

Примеры использования:

```
static gis_core_connection_t connection;
gis_core_link_init( &connection );
```

Классификация:

ПК ЦКИ для ЗОСРВ «Нейтрино»

Тематические ссылки:

[gis_core_link_init\(\)](#), [gis_core_link_connect\(\)](#),

gis_core_driver_id_t

Уникальный идентификатор драйвера источника данных.

Формат:

```
#include <gis/gis_types.h>

typedef enum {
    GIS_CORE_DRIVERS_ALL          = -2,
    GIS_CORE_DRIVERS_NONE        = -1,
    GIS_CORE_DRIVERS_S57_LOCAL   = 0,
    GIS_CORE_DRIVERS_SXF_LOCAL,
    GIS_CORE_DRIVERS_MTW_LOCAL,
    GIS_CORE_DRIVERS_SHP_LOCAL,
    GIS_CORE_DRIVERS_GEOTIFF_LOCAL,
    GIS_CORE_DRIVERS_JPEG2000_LOCAL,
    GIS_CORE_DRIVERS_DTED_LOCAL,
    GIS_CORE_DRIVERS_PNG_LOCAL,
    GIS_CORE_DRIVERS_RSW_LOCAL,
    GIS_CORE_DRIVERS_USER_V_LOCAL,
    GIS_CORE_DRIVERS_USER_R_LOCAL,
    GIS_CORE_DRIVERS_END_ID,
} gis_core_driver_id_t;
```

Описание:

Перечисление `gis_core_driver_id_t` позволяет адресовать драйверы источников данных в запросах к картографическому ядру (сервису). Перечисление включает следующие определения источников картографических данных:

- `GIS_CORE_DRIVERS_ALL` — подразумевается использование всех драйверов источников данных
- `GIS_CORE_DRIVERS_NONE` — идентификатор драйвера источника данных не определен
- `GIS_CORE_DRIVERS_S57_LOCAL` — локальное хранилище карт в формате S-57
- `GIS_CORE_DRIVERS_SXF_LOCAL` — локальное хранилище карт в формате SXF
- `GIS_CORE_DRIVERS_MTW_LOCAL` — локальное хранилище карт в формате MTW
- `GIS_CORE_DRIVERS_SHP_LOCAL` — локальное хранилище карт в формате Shapefile
- `GIS_CORE_DRIVERS_GEOTIFF_LOCAL` - локальное хранилище электронных карт в формате GeoTIFF
- `GIS_CORE_DRIVERS_JPEG2000_LOCAL` - локальное хранилище электронных карт в формате JPEG2000
- `GIS_CORE_DRIVERS_DTED_LOCAL` - локальное хранилище электронных карт в формате DTED

- GIS_CORE_DRIVERS_PNG_LOCAL - локальное хранилище электронных карт в формате PNG
- GIS_CORE_DRIVERS_RSW_LOCAL - локальное хранилище электронных карт в формате RSW
- GIS_CORE_DRIVERS_USER_V_LOCAL - локальное хранилище карт польз. векторного формата
- GIS_CORE_DRIVERS_USER_R_LOCAL - локальное хранилище карт польз. растрового формата

Параметр *GIS_CORE_DRIVERS_KKS* считается устаревшим, начиная с ПК ЦКИ редакции 1.3

Параметр *GIS_CORE_DRIVERS_UKS* считается устаревшим, начиная с ПК ЦКИ редакции 1.3

Параметр *GIS_CORE_DRIVERS_RER* считается устаревшим, начиная с ПК ЦКИ редакции 1.3

Классификация:

ПК ЦКИ для ЗОСРВ «Нейтрино»

gis_core_driver_info_t

Структура информации о драйвере.

Формат:

```
#include <gis/gishelper.h>

typedef struct
{
    gis_core_driver_id_t driver_id;
    bool                initialized;
    char                orig_fpath[GIS_MAX_PATH_LENGTH];
    union
    {
        struct {
            char    sxf_dirpath[GIS_MAX_PATH_LENGTH];
        } rer_data;
    } formats;
} gis_core_driver_info_t;
```

Описание:

Структура `gis_core_driver_info_t` описывает структуру, содержащую информацию о драйвере, поля интерпретируются следующим образом:

Аргументы:

driver_id

Тип драйвера ([gis_core_driver_id_t](#)).

initialized

Флаг инициализации.

orig_fpath

Полный путь к директории драйвера.

sxf_dirpath

Полный путь к директории "sxf".

Классификация:

ПК ЦКИ для ЗОСРВ «Нейтрино»

Тематические ссылки:

[gis_core_map_data_source_t](#), [gis_core_driver_id_t](#),

gis_core_hash_t

Хэш файла или карты.

Описание:

Тип данных позволяет хранить хэш произвольного файла и чаще всего используется для контроля целостности ЦКИ. Точное определение типа зависит от параметров окружения текущей версии картографического пакета.

Классификация:

ПК ЦКИ для ЗОСРВ «Нейтрино»

gis_core_map_data_source_t

Тип оригинала карты во внутреннем формате (GCM).

Формат:

```
#include <gis/gis_types.h>

typedef struct {
    GIS_CORE_MAP_DATA_SOURCE_ALL    = -2,
    GIS_CORE_MAP_DATA_SOURCE_NONE  = -1,
    GIS_CORE_MAP_DATA_SOURCE_S57   = 0,
    GIS_CORE_MAP_DATA_SOURCE_SXF
    GIS_CORE_MAP_DATA_SOURCE_MTW
    GIS_CORE_MAP_DATA_SOURCE_SHP
    GIS_CORE_MAP_DATA_SOURCE_GEOTIFF
    GIS_CORE_MAP_DATA_SOURCE_JPEG2000
    GIS_CORE_MAP_DATA_SOURCE_DTED
    GIS_CORE_MAP_DATA_SOURCE_PNG
    GIS_CORE_MAP_DATA_SOURCE_RSW
    GIS_CORE_MAP_DATA_SOURCE_END
} gis_core_map_data_source_t;
```

Описание:

Перечисление `gis_core_map_data_source_t` позволяет адресовать источники картографической информации, вернее, типы оригиналов карт из которых картографическим ядром (сервисом) воссозданы карты во внутреннем формате (GCM). Перечисление включает следующие определения оригинальных форматов карт:

- `GIS_CORE_MAP_DATA_SOURCE_ALL` — электронные карты всех перечисленных форматов
- `GIS_CORE_MAP_DATA_SOURCE_S57` — электронные карты в формате S-57
- `GIS_CORE_MAP_DATA_SOURCE_SXF` — электронные карты в формате SXF
- `GIS_CORE_MAP_DATA_SOURCE_MTW` — электронные карты в формате MTW
- `GIS_CORE_MAP_DATA_SOURCE_SHP` — электронные карты в формате Shapefile
- `GIS_CORE_MAP_DATA_SOURCE_GEOTIFF` — электронные карты в формате GeoTIFF
- `GIS_CORE_MAP_DATA_SOURCE_JPEG2000` — электронные карты в формате JPEG2000
- `GIS_CORE_MAP_DATA_SOURCE_DTED` — электронные карты в формате DTED
- `GIS_CORE_MAP_DATA_SOURCE_PNG` — электронные карты в формате PNG
- `GIS_CORE_MAP_DATA_SOURCE_RSW` — электронные карты в формате RSW
- `GIS_CORE_MAP_DATA_SOURCE_NONE` — для внутреннего использования
- `GIS_CORE_MAP_DATA_SOURCE_END` — для внутреннего использования

Классификация:

ПК ЦКИ для ЗОСРВ «Нейтрино»

gis_core_map_information_t

Информация о карте во внутреннем формате (GCM).

Формат:

```
#include <gis/gishelper.h>

typedef struct {
    int32_t            id;
    uint32_t           scale_denominator;
    char               gcm_fname[GIS_MAX_NAME_LENGTH];
    char               src_fname[GIS_MAX_NAME_LENGTH];
    char               src_bname[GIS_MAX_NAME_LENGTH];
    char**             gcm_clip_fnames;
    uint32_t           gcm_clip_count;
    gis_borders_t      borders;
    uint32_t           class_count;
    uint32_t           object_count;
    gis_core_map_data_source_t data_source;
    gis_core_driver_id_t driver_id;
    gis_map_projection_t meters_projection;
    gis_core_hash_t     orig_file_hash;
    union {
        struct {
            char               classifier_filename[GIS_MAX_NAME_LENGTH];
            gis_core_hash_t     classifier_file_hash;
        } __packed           vector;
        struct {
            double             min_height;
            double             max_height;
        } __packed           raster;
    } __packed              formats;
} gis_core_map_information_t;
```

Описание:

Структура `gis_core_map_information_t` описывает информацию о карте, поля интерпретируются следующим образом:

Аргументы:

id

Уникальный идентификатор карты в пределах драйвера.

scale_denominator

Знаменатель масштаба

gcm_fname

Имя файла карты во внутреннем формате (массив символов размером GIS_MAX_MAP_NAME_LENGTH).

src_fname

Имя исходного файла карты (массив символов размером GIS_MAX_MAP_NAME_LENGTH).

src_bname

Имя исходного файла карты без расширения (массив символов размером GIS_MAX_MAP_NAME_LENGTH).

src_version

Версия исходного файла карты (массив символов размером GIS_MAX_MAP_NAME_LENGTH).

gcm_clip_fnames

Массив имен нарезанных карт

gcm_clip_count

Количество нарезанных карт

borders

Границы карты в градусах (структура типа [gis_borders_t](#)).

class_count

Количество уникальных классов объектов.

object_count

Общее количество объектов в карте.

data_source

Тип оригинала данной карты ([gis_core_map_data_source_t](#)).

driver_id

Идентификатор драйвера картографического ядра ([gis_core_driver_id_t](#)).

meters_projection

Проекция карты ([gis_map_projection_t](#)).

orig_file_hash

Контрольная сумма оригинала карты ([gis_core_hash_t](#)).

Специфичные для формата оригинала GIS_CORE_MAP_DATA_SOURCE_VECTOR поля: **Аргументы:**

classifier_filename

Имя файла классификатора.

classifier_file_hash

Контрольная сумма файла классификатора.

Специфичные для формата оригинала GIS_CORE_MAP_DATA_SOURCE_MTW поля: **Аргументы:**

min_height

Нижний предел высот карты в метрах.

max_height

Верхний предел высот карты в метрах.

Классификация:

ПК ЦКИ для ЗОСРВ «Нейтрино»



Поле `src_version` доступно с версии 1.4

Тематические ссылки:

`gis_core_map_list_t`, `gis_borders_t`, `gis_core_map_data_source_t`, `gis_core_hash_t`, `gis_core_driver_id_t`,

gis_core_map_list_t

Список карт, доступных для выбранного драйвера.

Формат:

```
#include <gis/gishelper.h>

typedef struct {
    gis_core_map_information_t *entry_list;
    uint32_t entry_count;
} gis_core_map_list_t;
```

Описание:

Структура [gis_core_map_information_t](#) описывает список карт во внутреннем формате (GCM), поля интерпретируются следующим образом:

Аргументы:

entry_list

Динамический массив записей обо всех доступных картах ([gis_core_map_information_t](#)).

entry_count

Количество карт

Классификация:

ПК ЦКИ для ЗОСРВ «Нейтрино»

Тематические ссылки:

[gis_core_map_information_t](#), [gis_core_request_maps_list\(\)](#), [gis_core_map_list_free\(\)](#),

gis_core_request_parameters_t

Структура запроса к ядру картографического сервиса.

Формат:

```
#include <gis/gishelper.h>

typedef struct gis_core_request_parameters {
    gis_borders_t          borders;
    uint32_t               scale_lower_limit;
    uint32_t               scale_upper_limit;
    gis_core_map_data_source_t format;
    gis_core_driver_id_t   driver_id;
    uint16_t               cell_size;
} gis_core_request_parameters_t;
```

Описание:

Структура `gis_core_request_parameters_t` описывает структуру запроса к ядру, поля интерпретируются следующим образом:

Аргументы:

borders

Границы региона запроса (`gis_borders_t`).

scale_lower_limit

Нижний предел знаменателя масштаба карт.

scale_upper_limit

Верхний предел знаменателя масштаба карт.

format

Тип запрашиваемых данных (`gis_core_map_data_source_t`).

driver_id

Тип драйвера (`gis_core_driver_id_t`).

cell_size

Размер ячейки (специфичное поле для драйвера RER).

Классификация:

ПК ЦКИ для ЗОСРВ «Нейтрино»

Тематические ссылки:

`gis_borders_t`, `gis_core_map_data_source_t`, `gis_core_driver_id_t`,

gis_core_update_cache_mode_t

Режимы обновления кэша данных.

Формат:

```
#include <gis/gishelper.h>

typedef enum {
    GIS_CORE_UPDATE_CACHE_MODE_HARD_RESET,
    GIS_CORE_UPDATE_CACHE_MODE_SYNC_UPDATE,
    GIS_CORE_UPDATE_CACHE_MODE_SOFT_UPDATE
} gis_core_update_cache_mode_t;
```

Описание:

Перечисление gis_core_update_cache_mode_t включают следующие значения:

- GIS_CORE_UPDATE_CACHE_MODE_HARD_RESET — режим полного удаления карт GCM и создания новых из оригинального источника
- GIS_CORE_UPDATE_CACHE_MODE_SYNC_UPDATE — режим синхронизации карт GCM с оригинальным источником
- GIS_CORE_UPDATE_CACHE_MODE_SOFT_UPDATE — режим обновления метаинформации GCM без обновления карт во внутреннем формате

Сравнение режимов обновления:

Операция	Режим (1)	Режим (2)	Режим (3)
Удаление всех преобразованных во внутренний формат карт	+	-	-
Поиск новых карт в исходном формате и их конвертирование во внутренний формат	+	+	-
Поиск карт с измененными контрольными суммами в исходном формате и их конвертирование во внутренний формат	-	+	-

Поиск удаленных карт в исходном формате и удаление их преобразованных копий	-	+	-
Удаление неактуальных карт в разделяемой памяти	+	+	+
Удаление всех хранящихся ядром сведений о неактуальных картах	+	+	+
Загрузка в картографическое ядро краткой идентификационной информации карт	+	+	+
Загрузка в разделяемую память измененных карт	-	-	-

Классификация:

ПК ЦКИ для ЗОСРВ «Нейтрино»

Тематические ссылки:

[gis_core_request_parameters_t](#)

GIS_CORE_DRIVER_FOR_LOOP_HEAD

Обход всех доступных драйверов в цикле.

Формат:

```
#include <gis/gis_types.h>

#define GIS_CORE_DRIVER_FOR_LOOP_HEAD( iter )    for ( iter = GIS_CORE_DRIVERS_START_ID;
↪ iter < GIS_CORE_DRIVERS_END_ID; iter++ )
```

Аргументы:

iter

Итератор для драйверов источников данных. См. [gis_core_driver_id_t](#).

Библиотека:

gishelper

Описание:

Макрос позволяет проходить по всем доступным драйверам карт в цикле и выполнять над каждым из них какие-либо действия.

Примеры использования:

```
gis_core_driver_id_t driver_id;
char acronym[GIS_MAX_NAME_LENGTH];
gis_core_connection_t connection;

gis_core_link_init( &connection );
gis_core_link_connect( &connection, 777 );

GIS_CORE_DRIVER_FOR_LOOP_HEAD( driver_id )
{
    gis_core_request_driver_acronym( &connection, driver_id, &acronym );
    printf( "Acronym for driver %d is %s!\n", driver_id, &acronym[0] );
}
```

Тематические ссылки:

[gis_core_driver_id_t](#)

Классификация:

ПК ЦКИ для ЗОСРВ «Нейтрино»

GIS_CORE_DRIVER_ID_CHECK

Контроль корректности идентификатора драйвера.

Формат:

```
#include <gis/gis_types.h>

#define GIS_CORE_DRIVER_ID_CHECK( id )          (id >= GIS_CORE_DRIVERS_START_ID && id <
↪ GIS_CORE_DRIVERS_END_ID)
```

Аргументы:

id

Идентификатор драйвера источника данных. См. [gis_core_driver_id_t](#).

Библиотека:

gishelper

Описание:

Макрос позволяет проверить, существует ли драйвер с идентификатором, переданным в качестве параметра.

Возвращаемое значение:

0

Некорректный драйвер.

1

Корректный драйвер.

Примеры использования:

```
gis_core_driver_id_t driver_id = GIS_CORE_DRIVERS_NONE;

if ( !GIS_CORE_DRIVER_ID_CHECK( driver_id ) )
{
    printf( "Incorrect driver!\n" );
}
```

Тематические ссылки:

[gis_core_driver_id_t](#)

Классификация:

ПК ЦКИ для ЗОСРВ «Нейтрино»

GIS_IS_CORE_MAP_DATA_SOURCE_VECTOR

Проверка, является ли выбранный источник данных векторным либо растровым.

Формат:

```
#include <gis/gis_types.h>

#define GIS_CORE_DRIVER_ID_CHECK( id )          (id >= GIS_CORE_DRIVERS_START_ID && id <
↪ GIS_CORE_DRIVERS_END_ID)
```

Аргументы:

id

Источник данных. См. [gis_core_map_data_source_t](#).

Библиотека:

gishelper

Описание:

Макрос позволяет проверить, является ли источник данных, переданный в качестве параметра, векторным. В случае векторного источника данных, возвращается 1, в случае растрового - 0. Проверка выполняется путём наложения соответствующей маски (операция &) на передаваемый параметр. Для пользователя доступны следующие маски:

- GIS_CORE_MAP_DATA_SOURCE_VECTOR
- GIS_CORE_MAP_DATA_SOURCE_RASTER

Возвращаемое значение:

0

Растровый источник данных.

1

Векторный источник данных.

Примеры использования:

```
gis_core_map_data_source_t data_source = GIS_CORE_MAP_DATA_SOURCE_GEO TIFF;

if ( GIS_IS_CORE_MAP_DATA_SOURCE_VECTOR( data_source ) )
{
    printf( "This is vector data source!\n" );
}
else
{
    printf( "This is raster data source!\n" );
}
```

Тематические ссылки:

[gis_core_map_data_source_t](#)

Классификация:

ПК ЦКИ для ЗОСРВ «Нейтрино»

gis_helper_get_map_style_by_data_source()

Получение типа стиля карты по формату карты.

Прототип:

```
#include <gis/gis_types.h>

static inline gis_map_style_t gis_helper_get_map_style_by_data_source(
↪ gis_core_map_data_source_t src );
```

Аргументы:

data_source

Индекс типа данных карты.

Библиотека:

gishelper

Описание:

Функция возвращает тип стиля карты, соответствующий типу данных карты.

Возвращаемое значение:

!GIS_MAP_STYLE_END

Типа оригинала карты

GIS_MAP_STYLE_END

Задан некорректный идентификатор типа данных карты

Примеры использования:

```
gis_core_map_data_source_t src = GIS_CORE_MAP_DATA_SOURCE_S57;

if ( gis_helper_get_map_style_by_data_source( src ) != GIS_MAP_STYLE_S52 )
{
    printf( "Map style is invalid\n" );
    return 1;
}
```

Классификация:

Безопасность использования	
Обработчик прерываний	Да
Обработчик сигналов	Да
В потоке	Да

Тематические ссылки:

[gis_core_map_data_source_t](#), [gis_map_style_t](#)

gis_helper_get_map_style_name()

Получение строки с названием типа стиля карты.

Прототип:

```
#include <gis/gis_types.h>

static inline const char * gis_helper_get_map_style_name( gis_map_style_t style );
```

Аргументы:

style

Индекс типа стиля карты.

Библиотека:

gishelper

Описание:

Функция возвращает статическую строку с названием типа стиля отображения карты.

Возвращаемое значение:

!NULL

Строка с названием типа оригинала карты

NULL

Передан некорректный style

Примеры использования:

```
gis_map_style_t style = GIS_MAP_STYLE_RSC;

if ( strcmp (gis_helper_get_map_style_name( style ), "RSC" ) != 0 )
{
    printf( "Incorrect string: %s\n", gis_helper_get_map_style_name( style ) );
    return 1;
}
```

Классификация:

Безопасность использования	
Обработчик прерываний	Нет
Обработчик сигналов	Нет
В потоке	Да

Тематические ссылки:

[gis_map_style_t](#)

gis_helper_is_map_style_valid()

Проверка корректности типа стиля карты.

Прототип:

```
#include <gis/gis_types.h>

static inline bool gis_helper_is_map_style_valid( gis_map_style_t style );
```

Аргументы:

style

Индекс типа стиля карты.

Библиотека:

gishelper

Описание:

Функция проверяет типа оригинала карты на корректность.

Возвращаемое значение:

true

Тип оригинала карты заполнен успешно

false

Тип оригинала карты не доступен для использования

Примеры использования:

```
gis_map_style_t style = GIS_MAP_STYLE_S52;

if ( !gis_helper_is_map_style_valid( style ) )
{
    printf( "Map style is invalid\n" );
    return 1;
}
```

Классификация:

Безопасность использования	
Обработчик прерываний	Да
Обработчик сигналов	Да
В потоке	Да

Тематические ссылки:

[gis_map_style_t](#)

gis_map_style_fill_active_classifier_name_list()

Получение списка классификаторов отображаемых карт.

Прототип:

```
#include <gis/gishelper.h>

char **gis_map_style_fill_active_classifier_name_list( gis_borders_t *region );
```

Аргументы:

region

Границы холста типа `gis_borders_t`

Библиотека:

gishelper

Описание:

Функция ищет активные карты, которые имеют пересечение с границами `region`, и возвращает список классификаторов найденных карт. Список не содержит дубликаты.

Возвращаемое значение:

nullptr

Функция завершилась с ошибкой.



Функция выделяет память для передаваемого списка строк. Для освобождения памяти исп. `gis_helper_free_string_list()`.

Примеры использования:

```
gis_borders_t borders;
borders.degrees.west  = 180;
borders.degrees.east  = 180;
borders.degrees.north = 90;
borders.degrees.south = 90;

char **classifier_list = gis_map_style_fill_active_classifier_name_list( &borders );

if ( !classifier_list )
{
    return 1;
}

gis_helper_free_string_list( &classifier_list );
return 0;
```

Классификация:

ПК ЦКИ для ЗОСРВ «Нейтрино»

Безопасность использования	
Обработчик прерываний	Нет
Обработчик сигналов	Да
В потоке	Да

Поддерживается, начиная с ПК ЦКИ редакции 1.4

Тематические ссылки:

[gis_helper_free_string_list\(\)](#)

gis_map_style_flip_language()

Переключение языка карты на международный/национальный.

Прототип:

```
#include <gis/gishelper.h>

int gis_map_style_flip_language( gis_map_style_t map_style );
```

Аргументы:

map_style

Стиль векторной карты.

Библиотека:

gishelper

Описание:

Функция меняет значение флага национального языка. Возвращает EOK, если значение успешно изменено.



Функция поддерживает только карты, преобразованные из формата S-57.

Примеры использования:

```
gis_core_map_data_source_t map_data_source = GIS_CORE_MAP_DATA_SOURCE_S57;
gis_map_style_t map_style = gis_helper_get_map_style_by_data_source( map_data_source );

int success = gis_map_style_flip_language( map_style );

if ( success != EOK )
{
    printf( "Failed to set idx!\n" );
    return 1;
}
```

Классификация:

Безопасность использования	
Обработчик прерываний	Нет
Обработчик сигналов	Да
В потоке	Да

Поддерживается, начиная с ПК ЦКИ редакции 1.3

Тематические ссылки:

gis_map_style_t

gis_map_style_get_active_palette_idx()

Получение индекса активной палитры.

Прототип:

```
#include <gis/gishelper.h>

int gis_map_style_get_active_palette_idx( const gis_vector_palette_list_t *palette_v_list
↵ );
```

Аргументы:

palette_v_list

Список палитр для одного стиля векторных карт, полученный функцией [gis_map_style_get_palette_list\(\)](#).

Библиотека:

gishelper

Описание:

Функция возвращает значение индекса активной палитры для стиля карт переданного списка палитр. Возвращает -1 в случае неудачи.



Функция поддерживает только карты, преобразованные из формата S-57.

Примеры использования:

```
gis_core_map_data_source_t map_data_source = GIS_CORE_MAP_DATA_SOURCE_S57;
gis_map_style_t map_style = gis_helper_get_map_style_by_data_source( map_data_source );

gis_vector_palette_list_t *palettes = gis_map_style_get_palette_list( map_style );

if ( palettes == NULL )
{
    return 1;
}
else
{
    int idx = gis_map_style_get_active_palette_idx( palettes );
```

```

if ( idx != 0 ) // default value
{
    printf( "Incorrect idx!\n" );
    return 1;
}
gis_map_style_palette_list_free( &palettes );
}

```

Классификация:

ПК ЦКИ для ЗОСРВ «Нейтрино»

Безопасность использования	
Обработчик прерываний	Нет
Обработчик сигналов	Да
В потоке	Да

Поддерживается, начиная с ПК ЦКИ редакции 1.3

Тематические ссылки:

[gis_map_style_get_palette_list\(\)](#)

gis_map_style_get_palette_list()

Получение списка палитр для выбранного стиля векторных карт.

Прототип:

```
#include <gis/gishelper.h>

gis_vector_palette_list_t *gis_map_style_get_palette_list( gis_map_style_t map_style );
```

Аргументы:

map_style

Стиль карт, определяющий палитры, которые будут добавлены в итоговый список.

Библиотека:

gishelper

Описание:

Функция получает список `gis_vector_palette_list_t` палитр `gis_vector_palette_t`.



Функция поддерживает только карты, преобразованные из формата S-57.

Примеры использования:

```
gis_core_map_data_source_t map_data_source = GIS_CORE_MAP_DATA_SOURCE_S57;
gis_map_style_t map_style = gis_helper_get_map_style_by_data_source( map_data_source );

gis_vector_palette_list_t *palettes = gis_map_style_get_palette_list( map_style );

if ( palettes == NULL )
{
    return 1;
}

gis_map_style_palette_list_free( &palettes );
```

Классификация:

Безопасность использования	
Обработчик прерываний	Нет
Обработчик сигналов	Да
В потоке	Да

Поддерживается, начиная с ПК ЦКИ редакции 1.3

Тематические ссылки:

`gis_vector_palette_t`, `gis_vector_palette_list_t`

gis_map_style_get_SEABED()

Получение параметров отображения глубины.

Прототип:

```
#include <gis/gishelper.h>

int gis_map_style_get_SEABED( double *s57_SFD,
                             double *s57_SHC,
                             double *s57_SFC,
                             double *s57_DPC,
                             bool *s57_two_shades,
                             bool *s57_shallow_pattern );
```

Аргументы:

s57_SFD

SAFETY_DEPTH - безопасная глубина в метрах.

s57_SHC

SHALLOW_CONTOUR - контур глубины мелководья в метрах.

s57_SFC

SAFETY_CONTOUR - контур безопасной глубины в метрах.

s57_DPC

DEEP_CONTOUR - контур большой глубины в метрах.

s57_two_shades

TWO_SHADES - флаг использования двух или четырех цветов.

s57_shallow_pattern

SHALLOW_PATTERN - флаг использования дополнительного выделения опасной (No-GO) зоны.

Библиотека:

gishelper

Описание:

Функция присваивает соответствующие значения переменным указателей. Возвращает *ЕОК* в случае успешного завершения.

Возвращаемое значение:

ЕОК

Успешное завершение.

EINVAL

Некорректные параметры функции.



Функция поддерживается только картами, преобразованными из формата S-57. Функция не инициализирует передаваемые ей указатели!

Значения параметров по умолчанию: s57_SHC = 2; s57_SFC = 30; s57_SFD = 30; s57_DPC = 30; s57_two_shades = true; s57_shallow_pattern = false;

Примеры использования:

```
double *s57_SFD{new double};
double *s57_SHC{new double};
double *s57_SFC{new double};
double *s57_DPC{new double};
bool *s57_two_shades{new bool};
bool *s57_shallow_pattern{new bool};

int success = gis_map_style_get_SEABED( s57_SFD, s57_SHC, s57_SFC, s57_DPC,
                                         s57_two_shades, s57_shallow_pattern );

delete s57_SFD;
delete s57_SHC;
delete s57_SFC;
delete s57_DPC;
delete s57_two_shades;
delete s57_shallow_pattern;

if ( success != EOK )
{
    printf( "Failed to get parameters!\n" );
    return 1;
}
```

Классификация:

ПК ЦКИ для ЗОСРВ «Нейтрино»

Безопасность использования	
Обработчик прерываний	Нет
Обработчик сигналов	Да
В потоке	Да

Поддерживается, начиная с ПК ЦКИ редакции 1.4

Тематические ссылки:

`gis_map_style_set_SEABED()`

gis_map_style_palette_get_entry_count()

Получение размера списка палитр.

Прототип:

```
#include <gis/gishelper.h>

size_t gis_map_style_palette_get_entry_count( const gis_vector_palette_list_t
↪ *palette_v_list );
```

Аргументы:

palette_v_list

Список палитр для одного стиля векторных карт, полученный функцией [gis_map_style_get_palette_list\(\)](#).

Библиотека:

gishelper

Описание:

Функция возвращает размер списка палитр [gis_vector_palette_list_t](#).



Функция поддерживает только карты, преобразованные из формата S-57.

Примеры использования:

```
gis_core_map_data_source_t map_data_source = GIS_CORE_MAP_DATA_SOURCE_S57;
gis_map_style_t map_style = gis_helper_get_map_style_by_data_source( map_data_source );

gis_vector_palette_list_t *palettes = gis_map_style_get_palette_list( map_style );

if ( palettes == NULL )
{
    return 1;
}
else
{
    size_t palettes_size = gis_map_style_palette_get_entry_count( palettes );
```

```

if ( palettes_size != 5 )
{
    printf( "Incorrect palette size: %u! Expected: 5\n", palettes_size );
    return 1;
}
gis_map_style_palette_list_free( &palettes );
}

```

Классификация:

ПК ЦКИ для ЗОСРВ «Нейтрино»

Безопасность использования	
Обработчик прерываний	Нет
Обработчик сигналов	Да
В потоке	Да

Поддерживается, начиная с ПК ЦКИ редакции 1.3

Тематические ссылки:

[gis_vector_palette_list_t](#), [gis_map_style_get_palette_list\(\)](#)

gis_map_style_palette_get_entry_list()

Получение непосредственно списка палитр из структуры.

Прототип:

```
#include <gis/gishelper.h>

const gis_vector_palette_t *gis_map_style_palette_get_entry_list( const
↪ gis_vector_palette_list_t *palette_v_list );
```

Аргументы:

palette_v_list

Список палитр для одного стиля векторных карт, полученный функцией `gis_map_style_get_palette_list()`.

Библиотека:

gishelper

Описание:

Функция получает динамический список элементов типа `gis_vector_palette_t` из структуры `gis_vector_palette_list_t`. Элементы списка доступны только для чтения.



Функция поддерживает только карты, преобразованные из формата S-57.

Примеры использования:

```
gis_core_map_data_source_t map_data_source = GIS_CORE_MAP_DATA_SOURCE_S57;
gis_map_style_t map_style = gis_helper_get_map_style_by_data_source( map_data_source );

gis_vector_palette_list_t *palettes = gis_map_style_get_palette_list( map_style );

if ( palettes == NULL )
{
    return 1;
}
else
{
    const gis_vector_palette_t *palettes_entry_list =
↪ gis_map_style_palette_get_entry_list( palettes );
```

```

if ( strcmp( palettes_entry_list[0].name, "DAY_BRIGHT" ) != 0 )
{
    printf( "Got wrong palette from list!\n" );
    return 1;
}
gis_map_style_palette_list_free( &palettes );
}

```

Классификация:

ПК ЦКИ для ЗОСРВ «Нейтрино»

Безопасность использования	
Обработчик прерываний	Нет
Обработчик сигналов	Да
В потоке	Да

Поддерживается, начиная с ПК ЦКИ редакции 1.3

Тематические ссылки:

[gis_vector_palette_t](#), [gis_vector_palette_list_t](#), [gis_map_style_get_palette_list\(\)](#)

gis_map_style_palette_get_map_style()

Получение стиля векторных карт из списка палитр.

Прототип:

```
#include <gis/gishelper.h>

gis_map_style_t gis_map_style_palette_get_map_style( const gis_vector_palette_list_t
↪ *palette_v_list );
```

Аргументы:

palette_v_list

Список палитр для одного стиля векторных карт, полученный функцией `gis_map_style_get_palette_list()`.

Библиотека:

gishelper

Описание:

Функция получает стиль `gis_map_style_t` из `gis_vector_palette_list_t`.



Функция поддерживает только карты, преобразованные из формата S-57.

Примеры использования:

```
gis_core_map_data_source_t map_data_source = GIS_CORE_MAP_DATA_SOURCE_S57;
gis_map_style_t map_style = gis_helper_get_map_style_by_data_source( map_data_source );

gis_vector_palette_list_t *palettes = gis_map_style_get_palette_list( map_style );

if ( palettes == NULL )
{
    return 1;
}
else
{
    gis_map_style_t map_style = gis_map_style_palette_get_map_style( palettes );
}
```

```

if ( map_style != GIS_MAP_STYLE_S52 )
{
    printf( "Incorrect map style!\n" );
    return 1;
}
gis_map_style_palette_list_free( &palettes );
}

```

Классификация:

ПК ЦКИ для ЗОСРВ «Нейтрино»

Безопасность использования	
Обработчик прерываний	Нет
Обработчик сигналов	Да
В потоке	Да

Поддерживается, начиная с ПК ЦКИ редакции 1.3

Тематические ссылки:

[gis_map_style_t](#), [gis_vector_palette_list_t](#), [gis_map_style_get_palette_list\(\)](#)

gis_map_style_palette_list_free()

Освобождение списка палитр.

Прототип:

```
#include <gis/gishelper.h>

void gis_map_style_palette_list_free( gis_vector_palette_list_t **palette_v_list );
```

Аргументы:

palette_v_list

Список палитр для одного стиля векторных карт, полученный функцией [gis_map_style_get_palette_list\(\)](#).

Библиотека:

gishelper

Описание:

Функция освобождает память, задействованную для хранения списка палитр в формате [gis_vector_palette_list_t](#).



Функция поддерживает только карты, преобразованные из формата S-57.

Примеры использования:

```
gis_core_map_data_source_t map_data_source = GIS_CORE_MAP_DATA_SOURCE_S57;
gis_map_style_t map_style = gis_helper_get_map_style_by_data_source( map_data_source );

gis_vector_palette_list_t *palettes = gis_map_style_get_palette_list( map_style );

if ( palettes == NULL )
{
    return 1;
}

gis_map_style_palette_list_free( &palettes );
```

Классификация:

Безопасность использования	
Обработчик прерываний	Нет
Обработчик сигналов	Да
В потоке	Да

Поддерживается, начиная с ПК ЦКИ редакции 1.3

Тематические ссылки:

`gis_vector_palette_list_t`, `gis_map_style_get_palette_list()`

gis_map_style_set_active_palette_idx()

Установка индекса активной палитры.

Прототип:

```
#include <gis/gishelper.h>

int gis_map_style_set_active_palette_idx( const gis_vector_palette_list_t *palette_v_list,
↪ int idx );
```

Аргументы:

palette_v_list

Список палитр для одного стиля векторных карт, полученный функцией `gis_map_style_get_palette_list()`.

idx

Индекс активной палитры. Должен быть в пределах $0 \leq \text{idx} < \text{gis_map_style_palette_get_entry_count}()$.

Библиотека:

gishelper

Описание:

Функция устанавливает значение индекса активной палитры для стиля карт переданного списка палитр. Возвращает ЕОК, если индекс успешно установлен.



Функция поддерживает только карты, преобразованные из формата S-57.

Примеры использования:

```
gis_core_map_data_source_t map_data_source = GIS_CORE_MAP_DATA_SOURCE_S57;
gis_map_style_t map_style = gis_helper_get_map_style_by_data_source( map_data_source );

gis_vector_palette_list_t *palettes = gis_map_style_get_palette_list( map_style );

if ( palettes == NULL )
{
    return 1;
}
else
```

```

{
    int success = gis_map_style_set_active_palette_idx( palettes, 0 );

    if ( success != EOK )
    {
        printf( "Failed to set idx!\n" );
    }
    gis_map_style_palette_list_free( &palettes );
}

```

Классификация:

ПК ЦКИ для ЗОСРВ «Нейтрино»

Безопасность использования	
Обработчик прерываний	Нет
Обработчик сигналов	Да
В потоке	Да

Поддерживается, начиная с ПК ЦКИ редакции 1.3

Тематические ссылки:

`gis_map_style_get_palette_list()`, `gis_map_style_palette_get_entry_count()`

gis_map_style_set_SEABED()

Установка параметров отображения глубины.

Прототип:

```
#include <gis/gishelper.h>

void gis_map_style_set_SEABED( double s57_SFD,
                               double s57_SHC,
                               double s57_SFC,
                               double s57_DPC,
                               bool s57_two_shades,
                               bool s57_shallow_pattern );
```

Аргументы:

s57_SHC

SAFETY_DEPTH - безопасная глубина в метрах.

s57_SHC

SHALLOW_CONTOUR - контур глубины мелководья в метрах.

s57_SFC

SAFETY_CONTOUR - контур безопасной глубины в метрах.

s57_DPC

DEEP_CONTOUR - контур большой глубины в метрах.

s57_two_shades

TWO_SHADES - флаг использования двух или четырёх цветов.

s57_shallow_pattern

SHALLOW_PATTERN - флаг использования дополнительного выделения опасной (No-GO) зоны.

Библиотека:

gishelper

Описание:

Функция устанавливает значения соответствующих параметров.



Функция поддерживается только картами, преобразованными из формата S-57.

Примеры использования:

```
double shallow_contour    = 2;  
double safety_contour     = 30;  
double safety_depth       = 30;  
double deep_contour       = 30;  
bool two_color_depth      = true;  
bool shallow_pattern      = false;  
  
gis_map_style_set_SEABED( safety_depth,  
                           shallow_contour,  
                           safety_contour,  
                           deep_contour,  
                           two_color_depth,  
                           shallow_pattern );
```

Классификация:

ПК ЦКИ для ЗОСРВ «Нейтрино»

Безопасность использования	
Обработчик прерываний	Нет
Обработчик сигналов	Да
В потоке	Да

Поддерживается, начиная с ПК ЦКИ редакции 1.4

Тематические ссылки:

`gis_map_style_get_SEABED()`

gis_map_style_t

Тип стиля карты.

Формат:

```
#include <gis/gis_types.h>

typedef enum
{
    GIS_MAP_STYLE_ALL    = -2,
    GIS_MAP_STYLE_NONE   = -1,
    GIS_MAP_STYLE_S52    = 0,
    GIS_MAP_STYLE_RSC    = 1,
    GIS_MAP_STYLE_SLD    = 2,
    GIS_MAP_STYLE_END
} gis_map_style_t;
```

Описание:

Перечисление gis_map_style_t позволяет идентифицировать формат файла стиля отображения карты. Перечисление включает следующие стили:

- GIS_MAP_STYLE_ALL — стили отображения всех поддерживаемых форматов карт
- GIS_MAP_STYLE_S52 — стиль отображения S52 для карт формата S-57
- GIS_MAP_STYLE_RSC — классификатор RSC для карт формата SXF
- GIS_MAP_STYLE_SLD — стиль отображения для карт формата Shapefile
- GIS_MAP_STYLE_NONE — для внутреннего использования
- GIS_MAP_STYLE_END — для внутреннего использования

Классификация:

ПК ЦКИ для ЗОСРВ «Нейтрино»

gis_vector_palette_list_t

Список палитр стиля векторных карт.

Формат:

```
#include <gis/gis_types.h>

typedef struct gis_vector_palette_list gis_vector_palette_list_t;
```

Описание:

Структура содержит список палитр `gis_vector_palette_t` одного стиля. Для получения значений полей структуры необходимо использовать следующие функции:

- `gis_map_style_palette_get_map_style()`
- `gis_map_style_palette_get_entry_count()`
- `gis_map_style_palette_get_entry_list()`

Классификация:

ПК ЦКИ для ЗОСРВ «Нейтрино»

Поддерживается, начиная с ПК ЦКИ редакции 1.3

Тематические ссылки:

`gis_map_style_t`, `gis_vector_palette_t`, `gis_map_style_palette_get_map_style()`, `gis_map_style_palette_get_entry_count()`, `gis_map_style_palette_get_entry_list()`

gis_vector_palette_t

Дескриптор векторной палитры.

Формат:

```
#include <gis/gis_types.h>

typedef struct gis_vector_palette
{
    char      *name;
    size_t    size;
} gis_vector_palette_t;
```

Описание:

Структура описывает векторную палитру и включает следующие поля:

Аргументы:

name

Название палитры.

size

Размер палитры.

Классификация:

ПК ЦКИ для ЗОСРВ «Нейтрино»

Поддерживается, начиная с ПК ЦКИ редакции 1.3

Тематические ссылки:

[gis_vector_palette_list_t](#)

gis_core_databuffer_attach()

Функция подключения области данных к программе.

Прототип:

```
#include <gis/gis_databuffer.h>

int gis_core_databuffer_attach( gis_core_connection_t *connection );
```

Аргументы:

connection

Указатель на дескриптор соединения, тип [gis_core_connection_t](#)

Библиотека:

gishelper

Описание:

Функция производит подключение области данных к данному процессу для дальнейшей работы с ним.

Возвращаемое значение:

EOK

Успешное завершение

EPIPE

Соединение с картографическим ядром (сервисом) не установлено

ESRCH

Запрос к ядру на подключение не выполнен

EBADF

Разделяемый объект не существует

EFAULT

Разделяемый объект не подключен

ENOSYS

Выполнено подключение при отрицательном идентификаторе разделяемого объекта [gis_core_link_connect\(\)](#)

Примеры использования:

```

gis_core_connection_t connection;
gis_core_link_init( &connection );
gis_core_link_connect( &connection, 777 );

if ( gis_core_databuffer_attach( &connection ) != EOK ) {
    printf( "gis_core_databuffer_attach() failed\n" );
    return 1;
}

uint32_t size;
gis_core_databuffer_size( &connection, &size );
if ( size != 33554432 ) {
    printf( "Size of databuffer is: %d\n", size );
    return 1;
}

```

Классификация:

ПК ЦКИ для ЗОСРВ «Нейтрино»

Безопасность использования	
Обработчик прерываний	Нет
Обработчик сигналов	Нет
В потоке	Да

Тематические ссылки:

[gis_core_databuffer_detach\(\)](#), [gis_core_link_connect\(\)](#)

gis_core_databuffer_check()

Функция проверки области данных.

Прототип:

```
#include <gis/gis_databuffer.h>

int gis_core_databuffer_check( gis_core_connection_t *connection );
```

Аргументы:

connection

Указатель на дескриптор соединения, тип `gis_core_connection_t`

Библиотека:

gishelper

Описание:

Функция производит проверку области данных на подключение/валидность.

Возвращаемое значение:

EOK

Успешное завершение

EIO

Область данных отключена/невалидна

Примеры использования:

```
gis_core_connection_t connection;
gis_core_link_init( &connection );
gis_core_link_connect( &connection, 777 );

if ( gis_core_databuffer_attach( &connection ) != EOK ) {
    printf( "gis_core_databuffer_attach() failed\n" );
    return 1;
}

if ( gis_core_databuffer_check( &connection ) != EOK ) {
    printf( "gis_core_databuffer_check() failed\n" );
}
```

```
    return 1;  
}
```

Классификация:

ПК ЦКИ для ЗОСРВ «Нейтрино»

Безопасность использования	
Обработчик прерываний	Нет
Обработчик сигналов	Нет
В потоке	Да

Тематические ссылки:

`gis_core_databuffer_data_request()`

gis_core_databuffer_data_request()

Функция запроса данных у картографического ядра для выбранного региона.

Прототип:

```
#include <gis/gis_databuffer.h>

int gis_core_databuffer_data_request( gis_core_connection_t *connection,
↪ gis_core_request_parameters_t *map, int autosize );
```

Аргументы:

connection

Указатель на дескриптор соединения, тип `gis_core_connection_t`]

map

Модифицируемые параметры запроса к картографическому ядру (сервису), тип `gis_core_request_parameters_t`

autosize

Установка данного параметра в "1" включает режима автоматической модификации размера ПОП, предельное значение задается переменной окружения `GIS_CORE_DATABUFFER_SIZE_LIMIT`.

Библиотека:

gishelper

Описание:

Функция производит запрос к ядру на заполнение данными в соответствии с параметрами. Область избираемых карт определяется структурой `gis_core_request_parameters_t`

Возвращаемое значение:

EOK

Успешное завершение

EINVAL

Некорректные аргументы

EPIPE

Соединение с картографическим ядром (сервисом) не установлено

ESRCH

Запрос к ядру не выполнен

E_OVERFLOW

Заполнение области данных окончено из-за недостаточного объема области данных

ENOSYS

Выполнен запрос размера области данных при отрицательном идентификаторе разделяемого объекта `gis_core_link_connect()`]

EFAULT

Не удалось автоматически изменить размер области данных

Примеры использования:

```
int autosize = 1;
gis_core_request_parameters_t map;
gis_core_connection_t connection;
gis_core_link_init( &connection );
gis_core_link_connect( &connection, 777 );
gis_core_request_parameters_init( &map );

if ( gis_core_databuffer_attach( &connection ) != EOK ) {
    printf( "gis_core_databuffer_attach() failed\n" );
    return 1;
}

if ( gis_core_databuffer_data_request( &connection, &map, autosize ) != EOK ) {
    printf( "gis_core_databuffer_data_request() failed\n" );
    return 1;
}
```

Классификация:

ПК ЦКИ для ЗОСРВ «Нейтрино»

Безопасность использования	
Обработчик прерываний	Нет
Обработчик сигналов	Нет
В потоке	Нет

Тематические ссылки:

`gis_core_databuffer_check()`

gis_core_databuffer_detach()

Функция отключения области данных от программы.

Прототип:

```
#include <gis/gis_databuffer.h>

int gis_core_databuffer_detach();
```

Библиотека:

gishelper

Описание:

Функция производит отключение области данных от данного процесса.

Возвращаемое значение:

EOK

Успешное завершение

EFAULT

Отключение завершилось с ошибками

Примеры использования:

```
gis_core_connection_t connection;
gis_core_link_init( &connection );
gis_core_link_connect( &connection, 777 );

if ( gis_core_databuffer_attach( &connection ) != EOK ) {
    printf( "gis_core_databuffer_attach() failed\n" );
    return 1;
}

if ( gis_core_databuffer_detach() != EOK ) {
    printf( "gis_core_databuffer_detach() failed\n" );
    return 1;
}
```

Классификация:

Безопасность использования	
Обработчик прерываний	Нет
Обработчик сигналов	Нет
В потоке	Да

Тематические ссылки:

[gis_core_databuffer_attach\(\)](#), [gis_core_link_connect\(\)](#)

gis_core_databuffer_resize()

Задание размера области данных.

Прототип:

```
#include <gis/gis_databuffer.h>

int gis_core_databuffer_resize( gis_core_connection_t *connection, uint32_t size );
```

Аргументы:

connection

Указатель на дескриптор соединения, тип `gis_core_connection_t`

size

Значение нового размера области данных

Библиотека:

gishelper

Описание:

Функция использует соединение с картографическим ядром для запроса установки нового размера области данных.

Возвращаемое значение:

EOK

Успешное завершение

EINVAL

Некорректный аргумент

EPIPE

Соединение с картографическим ядром (сервисом) не установлено

EALREADY

Задан текущий размер области данных

ESRCH

Запрос установки размера области данных не выполнен

ENOMEM

Недостаточно оперативной памяти для установки нового размера РОП.

ENOSYS

Выполнен запрос размера области данных при отрицательном идентификаторе разделяемого объекта `gis_core_link_connect()`

EFAULT

Не удалось пересоздать область данных

Примеры использования:

```
uint32_t size;
gis_core_connection_t connection;
gis_core_link_init( &connection );
gis_core_link_connect( &connection, 777 );
gis_core_databuffer_attach( &connection );

gis_core_databuffer_size( &connection, &size );
printf( "Original size: %d\n", size );

int result;
uint32_t new_size = size + 10;
result = gis_core_databuffer_resize( &connection, new_size );
if ( result != EOK ) {
    printf( "gis_core_databuffer_resize() failed with %d\n", result );
    return 1;
}

gis_core_databuffer_size( &connection, &new_size );
printf( "New size: %d\n", new_size );
```

Классификация:

ПК ЦКИ для ЗОСРВ «Нейтрино»

Безопасность использования	
Обработчик прерываний	Нет
Обработчик сигналов	Нет
В потоке	Да

Тематические ссылки:

[gis_core_databuffer_size\(\)](#)

gis_core_databuffer_size()

Функция запроса размера области данных.

Прототип:

```
#include <gis/gis_databuffer.h>

int gis_core_databuffer_size( gis_core_connection_t *connection, uint32_t *size );
```

Аргументы:

connection

Указатель на дескриптор соединения, тип [gis_core_connection_t](#)

size

Указатель на переменную для хранения размера области данных

Библиотека:

gishelper

Описание:

Функция использует соединение с картографическим ядром для запроса размера области данных.

Возвращаемое значение:

EOK

Успешное завершение

EINVAL

Некорректный аргумент

EPIPE

Соединение с картографическим ядром (сервисом) не установлено

ESRCH

Запрос размера области данных не выполнен

ENOSYS

Выполнен запрос размера области данных при отрицательном идентификаторе разделяемого объекта [gis_core_link_connect\(\)](#)

Примеры использования:

```

uint32_t size;
gis_core_connection_t connection;
gis_core_link_init( &connection );
gis_core_link_connect( &connection, 777 );
gis_core_databuffer_attach( &connection );

if ( gis_core_databuffer_size( &connection, &size ) != EOK ) {
    printf( "gis_core_databuffer_size() failed\n" );
    return 1;
}

if ( size != 32412346 ) {
    printf( "Size of databuffer is: %d\n", size );
    return 1;
}

```

Классификация:

ПК ЦКИ для ЗОСРВ «Нейтрино»

Безопасность использования	
Обработчик прерываний	Нет
Обработчик сигналов	Нет
В потоке	Да

Тематические ссылки:

[gis_core_databuffer_resize\(\)](#)

gis_object_free()

Освобождение памяти объекта из области данных.

Прототип:

```
#include <gis/gis_objects.h>

void gis_object_free( gis_object_t *object );
```

Аргументы:

object

Указатель на структуру `gis_object_t`.

Библиотека:

gishelper

Описание:

Функция освобождает память объекта из области данных.

Примеры использования:

```
gis_object_t obj;
gis_object_init( &obj );
gis_object_free( &obj );

if ( obj.class_acronym != NULL ||
    obj.attributes      != NULL )
{
    printf( "Failed to free gis_object_t\n" );
    return 1;
}
```

Классификация:

ПК ЦКИ для ЗОСРВ «Нейтрино»

Безопасность использования	
Обработчик прерываний	Нет
Обработчик сигналов	Нет
В потоке	Да

Тематические ссылки:

`gis_object_t`, `gis_object_init()`

gis_object_init()

Инициализация объекта из области данных.

Прототип:

```
#include <gis/gis_objects.h>

void gis_object_init( gis_object_t *object );
```

Аргументы:

object

Указатель на структуру `gis_object_t`.

Библиотека:

gishelper

Описание:

Функция инициализирует объект из области данных.

Примеры использования:

```
gis_object_t obj;
gis_object_init( &obj );

if ( obj.class_acronym == NULL ||
    obj.attributes == NULL )
{
    printf( "Failed to initialize gis_object_t\n" );
    return 1;
}

gis_object_free( &obj );
```

Классификация:

Безопасность использования	
Обработчик прерываний	Нет
Обработчик сигналов	Нет
В потоке	Да

Тематические ссылки:

`gis_object_t`, `gis_object_free()`

gis_databuffer_desc_t

Дескриптор разделяемой области данных.

Формат:

```
#include <gis/gis_types.h>

typedef void *      gis_databuffer_desc_t;
```

Описание:

Дескриптор носит не публичный характер и определяет операции над разделяемой областью данных.

Классификация:

ПК ЦКИ для ЗОСРВ «Нейтрино»

class ClassifierInfo

Класс, описывающий информацию о конкретном классификаторе.

Формат:

```
#include <gis/gis_databuffer.h>

class ClassifierInfo
```

Описание:

Класс отображает информацию о конкретном классификаторе карты, а именно: индекс в РОП, имя классификатора, информацию о слоях.

Классификация:

ПК ЦКИ для ЗОСРВ «Нейтрино»

class ClassifierLayerInfo

Класс, описывающий информацию о конкретном слое в классификаторе.

Формат:

```
#include <gis/gis_databuffer.h>

class ClassifierLayerInfo
```

Описание:

Класс отображает информацию о конкретном слое выбранного классификатора карты, в том числе активность слоя.

Классификация:

ПК ЦКИ для ЗОСРВ «Нейтрино»

class DBCallbacks

Оповещение о состоянии разделяемой области памяти.

Формат:

```
#include <gis/gis_databuffer.h>

class DBCallbacks
```

Описание:

Класс предоставляет интерфейс оповещения о состоянии разделяемой области памяти. При изменении статуса РОП будет вызван `databufferValidityChangedCallback()`. Для того, чтобы использовать этот функционал, необходимо наследовать данный класс и реализовать метод. Публичные методы:

- `DBCallbacks::databufferValidityChangedCallback()` - функция обработчик изменения статуса РОП, содержащая флаг корректности РОП.

Классификация:

ПК ЦКИ для ЗОСРВ «Нейтрино»

class LayerInfo

Класс, описывающий информацию о слоях и поверхностях Surface Manager.

Формат:

```
#include <gis/gis_surfacemanager.h>

class LayerInfo
```

Описание:

Класс позволяет определить связку индекса поверхности и слоя.

Классификация:

ПК ЦКИ для ЗОСРВ «Нейтрино»

class MapInfo

Класс, описывающий информацию о карте в РОП.

Формат:

```
#include <gis/gis_databuffer.h>

class MapInfo
```

Описание:

Класс позволяет определить связку карты и её классификатора.

Классификация:

ПК ЦКИ для ЗОСРВ «Нейтрино»

class SMCallbacks

Класс, предоставляющий интерфейс функций-обработчиков для движка рендеринга Surface Manager.

Формат:

```
#include <gis/gis_surfacemanager.h>

class SMCallbacks
```

Описание:

Класс предоставляет интерфейс оповещения о состоянии движка рендеринга Surface Manager. Для того, чтобы использовать этот функционал, необходимо наследовать данный класс и реализовать метод. Публичные методы:

- `SMCallbacks::errorCallback()` — функция-обработчик ошибок рисования.
- `SMCallbacks::surfaceCompleteCallback()` — функция-обработчик события завершения рисования поверхности.
- `SMCallbacks::mapLayerCompleteCallback()` — функция-обработчик события завершения рисования слоя объектов векторной карты.
- `SMCallbacks::mapLayerRequiredCallback()` — функция-обработчик проверки активности слоя объектов векторной карты.
- `SMCallbacks::mapClassRequiredCallback()` — функция-обработчик проверки активности конкретного класса объектов векторной карты.

Классификация:

ПК ЦКИ для ЗОСРВ «Нейтрино»

DBCallbacks::databufferValidityChangedCallback()

Функция оповещения о корректности РОП.

Прототип:

```
#include <gis/gis_databuffer.h>

virtual void databufferValidityChangedCallback( bool isValid ) = 0;
```

Аргументы:

isValid

Флаг состояния РОП:

- 0 - разделяемая область памяти некорректна
- 1 - разделяемая область памяти корректна

Библиотека:

gishelper

Описание:

Функция предоставляет интерфейс оповещения об изменении состояния разделяемой области памяти.

Примеры использования:

```
class MapWidget : public DBCallbacks
public:
    void databufferValidityChangedCallback( bool isValid )
    {
        if ( isValid ) {
            printf( "Databuffer is valid \n" );
        } else {
            printf( "Databuffer is invalid \n" );
        }
    }
}
```

Классификация:

ПК ЦКИ для ЗОСРВ «Нейтрино»

Безопасность использования	
<i>Обработчик прерываний</i>	Нет
<i>Обработчик сигналов</i>	Нет
<i>В потоке</i>	Да

SMCallbacks::errorCallback()

Обработчик ошибок рисования.

Прототип:

```
#include <gis/gis_surfacemanager.h>

virtual void errorCallback( int err_code ) = 0;
```

Аргументы:

err_code

Код ошибки.

Библиотека:

surfacemanager

Описание:

Функция предоставляет интерфейс обработки ошибок рисования.

Примеры использования:

```
class MapWidget : public SMCallbacks
public:
    void errorCallback( int err_code )
    {
        switch (err_code) {
            case ETIME:
            {
                qDebug() << "Too long drawing.";
                break;
            }
            default:
                break;
        }
    }
}
```

Классификация:

Безопасность использования	
<i>Обработчик прерываний</i>	Нет
<i>Обработчик сигналов</i>	Нет
<i>В потоке</i>	Да

Поддерживается, начиная с ПК ЦКИ редакции 1.0.0

Поддерживается, начиная с ПК ЦКИ редакции 1.2

SMCallbacks::mapClassRequiredCallback()

Флаг необходимости рисования класса объектов векторной карты.

Прототип:

```
#include <gis/gis_surfacemanager.h>

virtual bool mapClassRequiredCallback( int32_t classifier_key, gis_core_class_code_t
↪ class_code ) = 0;
```

Аргументы:

classifier_key

Индекс классификатора.

class_code

Код класса.

Библиотека:

surfacemanager

Описание:

Функция предоставляет интерфейс включения/выключения рисования конкретного класса объектов векторной карты.

Примеры использования:

```
class MapWidget : public SMCallbacks
public:
    bool mapClassRequiredCallback( int32_t classifier_key, gis_core_class_code_t
↪ class_code )
    {
        return true;
    }
```

Классификация:

ПК ЦКИ для ЗОСРВ «Нейтрино»

Безопасность использования	
Обработчик прерываний	Нет
Обработчик сигналов	Нет
В потоке	Да

SMCallbacks::mapLayerCompleteCallback()

Завершение рисования слоя объектов векторной карты.

Прототип:

```
#include <gis/gis_surfacemanager.h>

virtual void mapLayerCompleteCallback( int32_t classifier_key, int32_t layer_number ) =
↪ 0;
```

Аргументы:

classifier_key

Индекс классификатора.

layer_number

Индекс слоя.

Библиотека:

surfacemanager

Описание:

Функция предоставляет интерфейс обработки окончания рисования слоя объектов векторной карты.

Примеры использования:

```
class MapWidget : public SMCallbacks
public:
    void mapLayerCompleteCallback( int32_t classifier_key, int32_t layer_number )
    {
        update();
    }
```

Классификация:

ПК ЦКИ для ЗОСРВ «Нейтрино»

Безопасность использования	
Обработчик прерываний	Нет
Обработчик сигналов	Нет
В потоке	Да

SMCallbacks::mapLayerRequiredCallback()

Флаг необходимости рисования слоя объектов векторной карты.

Прототип:

```
#include <gis/gis_surfacemanager.h>

virtual bool mapLayerRequiredCallback( int32_t classifier_key, int32_t layer_number ) =
↪ 0;
```

Аргументы:

classifier_key

Индекс классификатора.

layer_number

Индекс слоя.

Библиотека:

surfacemanager

Описание:

Функция предоставляет интерфейс включения/выключения рисования слоя объектов векторной карты.

Примеры использования:

```
class MapWidget : public SMCallbacks
{
public:
    bool mapLayerRequiredCallback( int32_t classifier_key, int32_t layer_number )
    {
        return true;
    }
};
```

Классификация:

ПК ЦКИ для ЗОСРВ «Нейтрино»

Безопасность использования	
Обработчик прерываний	Нет
Обработчик сигналов	Нет
В потоке	Да

SMCallbacks::surfaceCompleteCallback()

Завершение рисования поверхности.

Прототип:

```
#include <gis/gis_surfacemanager.h>

virtual void surfaceCompleteCallback( int32_t surface_z_idx ) = 0;
```

Аргументы:

surface_z_idx

Индекс поверхности.

Библиотека:

surfacemanager

Описание:

Функция предоставляет интерфейс обработки окончания рисования поверхности.

Примеры использования:

```
class MapWidget : public SMCallbacks
public:
    void surfaceCompleteCallback( int32_t surface_z_idx )
    {
        update();
    }
```

Классификация:

ПК ЦКИ для ЗОСРВ «Нейтрино»

Безопасность использования	
Обработчик прерываний	Нет
Обработчик сигналов	Нет
В потоке	Да

gis_object_t

Информация об объекте из области данных.

Формат:

```
#include <gis/gis_objects.h>

typedef struct
{
    gis_core_class_code_t    class_code;
    double                   height;
    gis_object_primitive_type_t type;
    uint32_t                 point_count;
    bool                     has_height;
    char*                    class_acronym;
    object_point_t*          points;
    char*                    attributes;
    gis_borders_t            bounding_rect;
} gis_object_t;
```

Описание:

Данный тип данных обеспечивает определение информации об объекте из области данных.

Аргументы:

class_code

Класс-код объекта. См. [gis_core_class_code_t](#).

type

Тип примитива объекта. См. [gis_object_primitive_type_t](#).

has_height

Флаг, указывающий, есть ли у объекта атрибут "высота". Если значение флага равно true, у объекта есть атрибут "высота". Значение атрибута записывается в поле height

height

Высота объекта. Указывается, если значение флага has_height равно true.

point_count

Количество точек объекта.

class_acronym

Название класса объекта

points

Указатель в область данных на массив точек объекта. Рекомендуется использовать только для чтения данных. См. [object_point_t](#).

attributes

Указатель на строку, содержащая атрибуты объекта.

bounding_rect

Координаты сторон описывающего прямоугольника для объекта. См. [gis_borders_t](#).

Классификация:

ПК ЦКИ для ЗОСРВ «Нейтрино»

Тематические ссылки:

[gis_core_class_code_t](#), [gis_object_primitive_type_t](#) [object_point_t](#) [gis_borders_t](#)

class GisObjectList

Список географических объектов.

Формат:

```
#include <gis/gis_objects.h>

class GisObjectList
```

Описание:

Класс осуществляет доступ к объектам из области данных. Класс позволяет фильтровать выбор объектов по класс-кодам и географическим границам. Публичные методы:

- `GisObjectList::GisObjectList()` — конструктор класса
- `GisObjectList::~GisObjectList()` — деструктор класса
- `GisObjectList::find_nearest_object()` — функция поиска ближайшего к заданной точке объекта
- `GisObjectList::find_nearest_point()` — функция поиска ближайшей точки объекта к заданной
- `GisObjectList::get_object()` — функция, возвращающая объект в списке по индексу
- `GisObjectList::get_object_count()` — функция, возвращающая количество объектов в списке

Классификация:

ПК ЦКИ для ЗОСРВ «Нейтрино»

Тематические ссылки:

`gis_borders_t`, `gis_object_t`, `object_point_t`, `gis_map_projection_t` `gis_core_class_code_t`

GisObjectList::find_nearest_object()

Функция поиска ближайшего к заданной точке объекта.

Прототип:

```
#include <gis/gis_objects.h>

int find_nearest_object( object_point_t point );
```

Аргументы:

point

Координаты точки

Библиотека:

gishelper

Описание:

Функция возвращает индекс ближайшего к заданной точке объекта в списке. Получить объект можно с помощью функции [GisObjectList::get_object\(\)](#)

Возвращаемое значение:

<

0 Найти ближайший объект не удалось.

>=

0 Индекс ближайшего к заданной точке объекта в списке.

Примеры использования:

```
void MapWidget::showObjectInfo(QPointF pos)
{
    ...
    object_point_t point_pos;
    ...
    int idx = list.find_nearest_object( point_pos );
    ...
}
```

Классификация:

ПК ЦКИ для ЗОСРВ «Нейтрино»

Безопасность использования	
<i>Обработчик прерываний</i>	Нет
<i>Обработчик сигналов</i>	Нет
<i>В потоке</i>	Да

Тематические ссылки:

`class GisObjectList, GisObjectList::get_object(), GisObjectList::find_nearest_point()`

GisObjectList::find_nearest_point()

Функция поиска ближайшей точки объекта к заданной.

Прототип:

```
#include <gis/gis_objects.h>

int find_nearest_point( object_point_t point, object_point_t *out );
```

Аргументы:

point

Координаты точки, ближайшую к которой требуется найти.

out

Указатель на структуру `object_point_t`, в которую записывается ближайшая точка.

Библиотека:

gishelper

Описание:

Функция находит точку, ближайшую к заданной точке среди всех точек объектов в списке.

Возвращаемое значение:

EOK

Ближайшая точка найдена

EINVAL

Не удалось найти ближайший объект

EFAULT

Не удалось найти ближайшую точку

Примеры использования:

```

void MapWidget::showObjectInfo(QPointF pos)
{
    ...
    object_point_t point_pos;
    ...
    object_point_t point_return;
    int status = list.find_nearest_point( point_pos, &point_return );
    ...
}

```

Классификация:

ПК ЦКИ для ЗОСРВ «Нейтрино»

Безопасность использования	
<i>Обработчик прерываний</i>	Нет
<i>Обработчик сигналов</i>	Нет
<i>В потоке</i>	Да

Тематические ссылки:

`class GisObjectList, GisObjectList::find_nearest_object()`

GisObjectList::get_object()

Функция, возвращающая объект в списке по индексу.

Прототип:

```
#include <gis/gis_objects.h>

int get_object( uint32_t idx, gis_object_t &object );
```

Аргументы:

idx

Индекс объекта в списке

object

Указатель на структуру `gis_object_t`, в которую записывается ближайшая точка.

Библиотека:

gishelper

Описание:

Функция заполняет переданный объект по индексу в списке `GisObjectList`.

Возвращаемое значение:

EOK

Объект найден

EINVAL

Индекс указан неверно

Примеры использования:

```
void MapWidget::showObjectInfo(QPointF pos)
{
    ...
    object_point_t point_pos;
    ...
    int idx = list.find_nearest_object( point_pos );
    gis_object_t _obj;
    list.get_object( idx, &_obj );
    ...
}
```

Классификация:

ПК ЦКИ для ЗОСРВ «Нейтрино»

Безопасность использования	
<i>Обработчик прерываний</i>	Нет
<i>Обработчик сигналов</i>	Нет
<i>В потоке</i>	Да

Тематические ссылки:

`class GisObjectList, GisObjectList::find_nearest_object()`

GisObjectList::get_object_count()

Функция, возвращающая количество объектов в списке.

Прототип:

```
#include <gis/gis_objects.h>

uint32_t get_object_count();
```

Описание:

Функция возвращает количество объектов в списке GisObjectList.

Возвращаемое значение:

uint32_t

Количество объектов в списке

Классификация:

ПК ЦКИ для ЗОСРВ «Нейтрино»

Безопасность использования	
Обработчик прерываний	Нет
Обработчик сигналов	Нет
В потоке	Да

Тематические ссылки:

[class GisObjectList](#), [GisObjectList::get_object\(\)](#)

GisObjectList:: GisObjectList()

Деструктор класса GisObjectList.

Прототип:

```
#include <gis/gis_objects.h>

~GisObjectList();
```

Библиотека:

gishelper

Описание:

Деструктор класса GisObjectList.

Классификация:

ПК ЦКИ для ЗОСРВ «Нейтрино»

Безопасность использования	
Обработчик прерываний	Нет
Обработчик сигналов	Нет
В потоке	Да

Тематические ссылки:

[class GisObjectList](#), [GisObjectList::GisObjectList\(\)](#)

GisObjectList::GisObjectList()

Конструктор класса *GisObjectList*.

Прототип:

```
#include <gis/gis_objects.h>
```

```
GisObjectList( gis_borders_t *region, std::vector<gis_core_class_code_t> class_list );
```

Аргументы:

region

Указатель на границы региона, внутри которого необходим выбор объектов. Тип [gis_borders_t](#)

class_list

Вектор класс-кодов, по которым необходим выбор объектов. Передача вектора с единственным значением класс-кода GIS_CLASS_CODE_UNDEFINED в конструктор GisObjectList осуществит выбор объектов любого класса.

classifier_name

Параметр необходим для фильтрации карт, внутри которых выбираются объекты, по классификатору.

Библиотека:

gishelper

Описание:

Конструктор инициализирует экземпляр класса GisObjectList, необходимый для получения объектов из области данных. Объекты фильтруются по границам региона и класс-кодам.

Примеры использования:

```
void MapWidget::showObjectInfo( QPointF pos )
{
    gis_core_request_parameters_t map;
    gis_core_request_parameters_init( &map );

    std::vector<gis_core_class_code_t> class_list;
    class_list.push_back( GIS_CLASS_CODE_UNDEFINED );
    GisObjectList list( &map.borders, class_list );
    ...
}
```

Классификация:

ПК ЦКИ для ЗОСРВ «Нейтрино»

Безопасность использования	
Обработчик прерываний	Нет
Обработчик сигналов	Нет
В потоке	Да



До версии 1.4 не поддерживалось использование classifier_name

Тематические ссылки:

`class GisObjectList, GisObjectList::get_object()`

gis_data_engine_alloc()

Создание контекста движка рендеринга.

Прототип:

```
#include <gis/gisrender.h>

int gis_data_engine_alloc( uint32_t width, uint32_t height, uint32_t mode, uint8_t bpp,
↳ gis_core_connection_t *connection, gis_data_engine_context_t *ctx );
```

Аргументы:

<i>width</i>	Ширина окна отображения.
<i>height</i>	Высота окна отображения.
<i>mode</i>	Режима работы движка рендеринга.
<i>bpp</i>	Глубина цвета буфера движка рендеринга.
<i>connection</i>	Указатель на дескриптор соединения с ядром.
<i>ctx</i>	Указатель на контекст движка рендеринга.

Библиотека:

gisrender

Описание:

Функция создаёт контекст движка рендеринга и инициализирует его переданными параметрами.

Возвращаемое значение:

<i>EOK</i>	Успешное завершение
<i>EINVAL</i>	Некорректные параметры функции
<i>ENOTSUP</i>	Некорректный режим движка рендеринга

ENOMEM

Недостаточно памяти

ENODATA

Разделяемая область памяти не подключена

Примеры использования:

```
gis_core_connection_t connection;
gis_data_engine_context_t ctx;
uint32_t width = 100;
uint32_t height = 100;

gis_core_link_init( &connection );
gis_core_link_connect( &connection, 777 );
gis_core_databuffer_attach( &connection );

if ( gis_data_engine_alloc( width, height, GIS_DATA_ENGINE_MODE_SM,
↪ GIS_DATA_ENGINE_BPP_32, &connection, &ctx ) != EOK ) {
    printf( "gis_data_engine_alloc() failed.\n" );
    return 1;
}

gis_data_engine_free( &ctx );
gis_core_databuffer_detach();
gis_core_link_destroy( &connection, true );
```

Классификация:

ПК ЦКИ для ЗОСРВ «Нейтрино»

Безопасность использования	
Обработчик прерываний	Нет
Обработчик сигналов	Нет
В потоке	Да

Тематические ссылки:

`gis_data_engine_context_t`, `gis_data_engine_update()`

gis_data_engine_free()

Освобождение контекста движка рендеринга.

Прототип:

```
#include <gis/gisrender.h>

void gis_data_engine_free( gis_data_engine_context_t *ctx );
```

Аргументы:

ctx

Указатель на контекст движка рендеринга

Библиотека:

gisrender

Описание:

Освобождение памяти, выделенной под контекст и внутренние поля движка рендеринга. Функция должна быть вызвана перед следующим вызовом `gis_data_engine_alloc()`.

Примеры использования:

```
gis_core_connection_t connection;
gis_data_engine_context_t ctx;
uint32_t width = 100;
uint32_t height = 100;

gis_core_link_init( &connection );
gis_core_link_connect( &connection, 777 );
gis_core_databuffer_attach( &connection );

if ( gis_data_engine_alloc( width, height, GIS_DATA_ENGINE_MODE_SM,
↳ GIS_DATA_ENGINE_BPP_32, &connection, &ctx ) != EOK ) {
    printf( "gis_data_engine_alloc() failed.\n" );
    return 1;
}

gis_data_engine_free( &ctx );
if ( ctx != NULL )
{
    printf( "gis_data_engine_free() failed.\n" );
}
```

```

        return 1;
    }

    gis_core_databuffer_detach();
    gis_core_link_destroy( &connection, true );

```

Классификация:

ПК ЦКИ для ЗОСРВ «Нейтрино»

Безопасность использования	
Обработчик прерываний	Нет
Обработчик сигналов	Нет
В потоке	Да

Тематические ссылки:

[gis_data_engine_context_t](#), [gis_data_engine_alloc\(\)](#)

gis_data_engine_get_canvas_size()

Получение размера окна отображения движка рендеринга.

Прототип:

```
#include <gis/gisrender.h>

int gis_data_engine_get_canvas_size( gis_data_engine_context_t ctx, uint32_t *width,
↪ uint32_t *height );
```

Аргументы:

<i>ctx</i>	Указатель на контекст движка рендеринга.
<i>width</i>	Указатель на ширину окна отображения.
<i>height</i>	Указатель на высоту окна отображения.

Библиотека:

gisrender

Описание:

Функция получает размеры окна отображения движка рендеринга.

Возвращаемое значение:

<i>EOK</i>	Успешное завершение
<i>EINVAL</i>	Некорректные параметры функции

Примеры использования:

```

gis_core_connection_t connection;
gis_data_engine_context_t ctx;
uint32_t width = 100;
uint32_t height = 100;

gis_core_link_init( &connection );
gis_core_link_connect( &connection, 777 );
gis_core_databuffer_attach( &connection );

if ( gis_data_engine_alloc( width, height, GIS_DATA_ENGINE_MODE_SM,
↪ GIS_DATA_ENGINE_BPP_32, &connection, &ctx ) != EOK ) {
    printf( "gis_data_engine_alloc() failed.\n" );
    return 1;
}

if ( gis_data_engine_set_canvas_size( ctx, 200, 100 ) != EOK ) {
    printf( "Failed to update canvas size \n" );
    return 1;
}

uint32_t test_width, test_height;
if ( gis_data_engine_get_canvas_size( ctx, &test_width, &test_height ) != EOK ) {
    printf( "Failed to get canvas size \n" );
    return 1;
}

if ( test_width != 200 && test_height != 100 ) {
    printf( "Incorrect canvas size \n" );
    return 1;
}

gis_data_engine_free( &ctx );
gis_core_databuffer_detach();
gis_core_link_destroy( &connection, true );

```

Классификация:

ПК ЦКИ для ЗОСРВ «Нейтрино»

Безопасность использования	
Обработчик прерываний	Нет
Обработчик сигналов	Нет
В потоке	Да

Поддерживается, начиная с ПК ЦКИ редакции 1.2

Тематические ссылки:

gis_data_engine_context_t, gis_data_engine_set_canvas_size() gis_data_engine_alloc(), gis_data_engine_update()

gis_data_engine_get_classifier_list()

Получение списка классификаторов с информацией о слоях.

Прототип:

```
#include <gis/gisrender.h>

int gis_data_engine_get_classifier_list( gis_data_engine_context_t ctx,
↪ std::vector<ClassifierInfo> &list );
```

Аргументы:

ctx
Указатель на контекст движка рендеринга.

list
Вектор, содержащий тип `class ClassifierInfo`.

Библиотека:

gisrender

Описание:

Функция заполняет список классификаторов, находящихся в области данных движка рендеринга.

Возвращаемое значение:

EOK
Успешное завершение

EINVAL
Некорректные параметры функции

ENODATA
Отсутствуют классификаторы

Примеры использования:

```

gis_core_connection_t connection;
gis_data_engine_context_t ctx;
uint32_t width = 100;
uint32_t height = 100;

gis_core_link_init( &connection );
if ( gis_core_link_connect( &connection, 777 ) != EOK ) {
    printf( "Failed to connect databuffer\n" );
    return 1;
}

if ( gis_core_databuffer_attach( &connection ) != EOK ) {
    printf( "Failed to attach databuffer\n" );
    return 1;
}

gis_core_request_parameters_t map;
gis_core_request_parameters_init( &map );
gis_core_databuffer_data_request( &connection, &map, 1 );

if ( gis_data_engine_alloc( width, height, GIS_DATA_ENGINE_MODE_SM,
↪ GIS_DATA_ENGINE_BPP_32, &connection, &ctx ) != EOK ) {
    printf( "gis_data_engine_alloc() failed.\n" );
    return 1;
}

std::vector <ClassifierInfo> list;
list.clear();

if ( gis_data_engine_get_classifier_list( ctx, list ) != EOK ) {
    printf( "Failed to fill classifier list\n" );
    return 1;
}

if ( list.size() != 1 )
{
    printf( "Incorrect classifier_list size: %d\n", list.size() );
    return 1;
}

gis_data_engine_free( &ctx );
gis_core_databuffer_detach();
gis_core_link_destroy( &connection, true );

```

Классификация:

ПК ЦКИ для ЗОСРВ «Нейтрино»

Безопасность использования	
<i>Обработчик прерываний</i>	Нет
<i>Обработчик сигналов</i>	Нет
<i>В потоке</i>	Да

Тематические ссылки:

`gis_data_engine_context_t`, `gis_data_engine_get_map_list()`

gis_data_engine_get_class_list()

Получение списка активных классов.

Прототип:

```
#include <gis/gisrender.h>

int gis_data_engine_get_class_list( gis_data_engine_context_t ctx,
↪ std::vector<ClassifierInfo> &list, std::vector<gis_core_class_code_t> &class_list );
```

Аргументы:

ctx

Указатель на контекст движка рендеринга.

list

Вектор, содержащий тип `class ClassifierInfo`.

class_list

Вектор активных кодов класса типа `gis_core_class_code_t`.

Библиотека:

gisrender

Описание:

Функция заполняет список классов, активных на момент запроса.

Возвращаемое значение:

EOK

Успешное завершение

EINVAL

Некорректные параметры функции

ENODATA

Отсутствуют классификаторы

Примеры использования:

```

gis_core_connection_t connection;
gis_data_engine_context_t ctx;
uint32_t width = 100;
uint32_t height = 100;

gis_core_link_init( &connection );
if ( gis_core_link_connect( &connection, 777 ) != EOK ) {
    printf( "Failed to connect databuffer\n" );
    return 1;
}

if ( gis_core_databuffer_attach( &connection ) != EOK ) {
    printf( "Failed to attach databuffer\n" );
    return 1;
}

gis_core_request_parameters_t map;
gis_core_request_parameters_init( &map );
gis_core_databuffer_data_request( &connection, &map, 1 );

if ( gis_data_engine_alloc( width, height, GIS_DATA_ENGINE_MODE_SM,
↪ GIS_DATA_ENGINE_BPP_32, &connection, &ctx ) != EOK ) {
    printf( "gis_data_engine_alloc() failed.\n" );
    return 1;
}

std::vector<ClassifierInfo> list;
std::vector<gis_core_class_code_t> class_list;
list.clear();
class_list.clear();

if ( gis_data_engine_get_classifier_list( ctx, list ) != EOK ) {
    printf( "Failed to fill classifier list \n" );
    return 1;
}

for ( int i = 0; i < (int)list.size(); i++ )
{
    printf( "classifier_%d: %s\n", i, list.at( i ).name.c_str() );
}

if ( gis_data_engine_get_class_list( ctx, list, class_list ) != EOK ) {
    printf( "Failed to fill class list\n" );
    return 1;
}

if ( class_list.size() != 1182 )
{
    printf( "Incorrect class_list size: %d\n", class_list.size() );
    return 1;
}

```

```

}

gis_data_engine_free( &ctx );
gis_core_databuffer_detach();
gis_core_link_destroy( &connection, true );

```

Классификация:

ПК ЦКИ для ЗОСРВ «Нейтрино»

Безопасность использования	
Обработчик прерываний	Нет
Обработчик сигналов	Нет
В потоке	Да

Поддерживается, начиная с ПК ЦКИ редакции 1.1.0

Тематические ссылки:

`gis_data_engine_context_t`, `gis_data_engine_get_map_list()`

gis_data_engine_get_display_parameters()

Получение указателя на контекст параметров визуализации.

Прототип:

```
#include <gis/gisrender.h>

gis_mdp_t gis_data_engine_get_display_parameters( gis_data_engine_context_t ctx );
```

Аргументы:

ctx

Указатель на контекст движка рендеринга

Библиотека:

gisrender

Описание:

Функция возвращает указатель на контекст параметров визуализации.

Возвращаемое значение:

!NULL

Успешное завершение.

NULL

Возникла ошибка или ctx указывает на NULL.

Примеры использования:

```
gis_core_connection_t connection;
gis_data_engine_context_t ctx;
uint32_t width = 100;
uint32_t height = 100;

gis_core_link_init( &connection );
gis_core_link_connect( &connection, 777 );
gis_core_databuffer_attach( &connection );

if ( gis_data_engine_alloc( width, height, GIS_DATA_ENGINE_MODE_SM,
↪ GIS_DATA_ENGINE_BPP_32, &connection, &ctx ) != EOK ) {
    printf( "gis_data_engine_alloc() failed.\n" );
}
```

```

        return 1;
    }

    gis_mdp_t disp_param = gis_data_engine_get_display_parameters( ctx );
    if ( !disp_param ) {
        printf( "gis_data_engine_get_display_parameters() failed\n" );
        return 1;
    }

    uint32_t scale = gis_mdp_get_phys_scale( disp_param );

    if ( scale != 1000000 )
    {
        printf( "Incorrect display parameters: %d\n", scale );
        return 1;
    }

    gis_data_engine_free( &ctx );
    gis_core_databuffer_detach();
    gis_core_link_destroy( &connection, true );

```

Классификация:

ПК ЦКИ для ЗОСРВ «Нейтрино»

Безопасность использования	
Обработчик прерываний	Нет
Обработчик сигналов	Да
В потоке	Да

Тематические ссылки:

[gis_data_engine_context_t](#), [gis_data_engine_alloc\(\)](#), [gis_data_engine_free\(\)](#)

gis_data_engine_get_map_list()

Получение списка карт с привязкой к классификаторам.

Прототип:

```
#include <gis/gisrender.h>

int gis_data_engine_get_map_list( gis_data_engine_context_t ctx, std::vector<MapInfo> &list
↵ );
```

Аргументы:

ctx
Указатель на контекст движка рендеринга.

list
Вектор, содержащий тип `class MapInfo`.

Библиотека:

gisrender

Описание:

Функция заполняет список карт, находящихся в области данных движка рендеринга.

Возвращаемое значение:

EOK
Успешное завершение

EINVAL
Некорректные параметры функции

ENODATA
Отсутствуют карты

Примеры использования:

```

gis_core_connection_t connection;
gis_data_engine_context_t ctx;
uint32_t width = 100;
uint32_t height = 100;

gis_core_link_init( &connection );
if ( gis_core_link_connect( &connection, 777 ) != EOK ) {
    printf( "Failed to connect databuffer\n" );
    return 1;
}

if ( gis_core_databuffer_attach( &connection ) != EOK ) {
    printf( "Failed to attach databuffer\n" );
    return 1;
}

gis_core_request_parameters_t map;
gis_core_request_parameters_init( &map );
gis_core_databuffer_data_request( &connection, &map, 1 );

if ( gis_data_engine_alloc( width, height, GIS_DATA_ENGINE_MODE_SM,
↪ GIS_DATA_ENGINE_BPP_32, &connection, &ctx ) != EOK ) {
    printf( "gis_data_engine_alloc() failed.\n" );
    return 1;
}

std::vector<MapInfo>    maps;

if ( gis_data_engine_get_map_list( ctx, maps ) != EOK ) {
    printf( "Failed to fill map list\n" );
    return 1;
}

if ( strcmp( maps[0].classifierName.c_str(), "200T-3.RSC.gcm" ) == 0 &&
    ( strcmp( maps[0].mapName.c_str(), "K3704.sxf" ) == 0 ||
      strcmp( maps[0].mapName.c_str(), "K3705.sxf" ) == 0 ) )
{
    printf( "Correct\n" );
}
else
{
    printf( "Incorrect map list\n" );
    printf( "Got first map: %s, expected: K3704.sxf || K3705.sxf\n",
                                                    maps[0].mapName.c_str() );
    return 1;
}

gis_data_engine_free( &ctx );
gis_core_databuffer_detach();
gis_core_link_destroy( &connection, true );

```

Классификация:

ПК ЦКИ для ЗОСРВ «Нейтрино»

Безопасность использования	
<i>Обработчик прерываний</i>	Нет
<i>Обработчик сигналов</i>	Нет
<i>В потоке</i>	Да

Тематические ссылки:

`gis_data_engine_context_t`, `gis_data_engine_get_classifier_list()`

gis_data_engine_get_maps_projection()

Получение информации о проекции отображения.

Прототип:

```
#include <gis/gisrender.h>

int gis_data_engine_get_maps_projection( gis_data_engine_context_t ctx, double_point_t
↪ *center_pnt, gis_map_projection_t *out_proj );
```

Аргументы:

<i>ctx</i>	Указатель на контекст движка рендеринга.
<i>center_pnt</i>	Указатель на центральную точку.
<i>out_proj</i>	Проекция карт.

Библиотека:

gisrender

Описание:

Функция позволяет получить информацию о проекции карты, которой принадлежит точка *center_pnt*.

Возвращаемое значение:

<i>EOK</i>	Успешное завершение
<i>EINVAL</i>	Некорректные параметры функции
<i>ENODATA</i>	Отсутствуют карты

Примеры использования:

```

gis_core_connection_t connection;
gis_data_engine_context_t ctx;
uint32_t width = 100;
uint32_t height = 100;

gis_core_link_init( &connection );
gis_core_link_connect( &connection, 777 );
gis_core_databuffer_attach( &connection );

if ( gis_data_engine_alloc( width, height, GIS_DATA_ENGINE_MODE_SM,
↪ GIS_DATA_ENGINE_BPP_32, &connection, &ctx ) != EOK ) {
    printf( "gis_data_engine_alloc() failed.\n" );
    return 1;
}

gis_map_projection_t proj_params;
double_point_t center;
center.x = 39.91;
center.y = 45.74;

gis_map_projection_init( &proj_params );

gis_data_engine_get_maps_projection( ctx, &center, &proj_params );

if ( proj_params.projection_idx == 5 &&
    proj_params.ellipsoid_idx == 3 &&
    proj_params.EPSG == 0 )
{
    printf( "Correct\n" );
}
else
{
    printf( "Incorrect parameters: %d %d %d\n", proj_params.projection_idx,
                                                proj_params.ellipsoid_idx,
                                                proj_params.EPSG );
}

gis_data_engine_free( &ctx );
gis_core_databuffer_detach();
gis_core_link_destroy( &connection, true );

```

Классификация:

ПК ЦКИ для ЗОСРВ «Нейтрино»

Безопасность использования	
<i>Обработчик прерываний</i>	Нет
<i>Обработчик сигналов</i>	Нет
<i>В потоке</i>	Да

Тематические ссылки:

`gis_data_engine_context_t`, `gis_map_projection_t`, `gis_data_engine_set_canvas_size()`

gis_data_engine_set_canvas_size()

Обновление размера окна отображения движка рендеринга.

Прототип:

```
#include <gis/gisrender.h>

int gis_data_engine_set_canvas_size( gis_data_engine_context_t ctx, uint32_t width,
↪ uint32_t height );
```

Аргументы:

<i>ctx</i>	Указатель на контекст движка рендеринга.
<i>width</i>	Ширина окна отображения.
<i>height</i>	Высота окна отображения.

Библиотека:

gisrender

Описание:

Функция обновляет размеры окна отображения движка рендеринга.

Возвращаемое значение:

<i>EOK</i>	Успешное завершение
<i>EINVAL</i>	Некорректные параметры функции

Примеры использования:

```

gis_core_connection_t connection;
gis_data_engine_context_t ctx;
uint32_t width = 100;
uint32_t height = 100;

gis_core_link_init( &connection );
gis_core_link_connect( &connection, 777 );
gis_core_databuffer_attach( &connection );

if ( gis_data_engine_alloc( width, height, GIS_DATA_ENGINE_MODE_SM,
↪ GIS_DATA_ENGINE_BPP_32, &connection, &ctx ) != EOK ) {
    printf( "gis_data_engine_alloc() failed.\n" );
    return 1;
}

if ( gis_data_engine_set_canvas_size( ctx, 200, 100 ) != EOK ) {
    printf( "Failed to update canvas size \n" );
    return 1;
}

uint32_t test_width, test_height;
if ( gis_data_engine_get_canvas_size( ctx, &test_width, &test_height ) != EOK ) {
    printf( "Failed to get canvas size \n" );
    return 1;
}

if ( test_width != 200 && test_height != 100 ) {
    printf( "Incorrect canvas size \n" );
    return 1;
}

gis_data_engine_free( &ctx );
gis_core_databuffer_detach();
gis_core_link_destroy( &connection, true );

```

Классификация:

ПК ЦКИ для ЗОСРВ «Нейтрино»

Безопасность использования	
Обработчик прерываний	Нет
Обработчик сигналов	Нет
В потоке	Да

Тематические ссылки:

[gis_data_engine_context_t](#), [gis_data_engine_get_canvas_size\(\)](#) [gis_data_engine_alloc\(\)](#), [gis_data_engine_update\(\)](#)

gis_data_engine_set_notify_func()

Установка функции-обработчика на событие изменения состояния области данных.

Прототип:

```
#include <gis/gisrender.h>

void gis_data_engine_set_notify_func( gis_data_engine_context_t *ctx, DBCallbacks
↪ *notify_class );
```

Аргументы:

ctx

Указатель на контекст движка рендеринга.

notify_class

Указатель на объект класса `class DBCallbacks`.

Библиотека:

gisrender

Описание:

Функция устанавливает функцию-обработчик на событие изменения состояния движка рендеринга.

Примеры использования:

```
MapWidget w;
gis_core_connection_t connection;
gis_data_engine_context_t ctx;
uint32_t width = 100;
uint32_t height = 100;

gis_core_link_init( &connection );
gis_core_link_connect( &connection, 777 );
gis_core_databuffer_attach( &connection );

if ( gis_data_engine_alloc( width, height, GIS_DATA_ENGINE_MODE_SM,
↪ GIS_DATA_ENGINE_BPP_32, &connection, &ctx ) != EOK ) {
    printf( "gis_data_engine_alloc() failed.\n" );
    return 1;
}
```

```
gis_data_engine_set_notify_func( &ctx, &w );

gis_data_engine_free( &ctx );
gis_core_databuffer_detach();
gis_core_link_destroy( &connection, true );
```

Классификация:

ПК ЦКИ для ЗОСРВ «Нейтрино»

Безопасность использования	
Обработчик прерываний	Нет
Обработчик сигналов	Нет
В потоке	Да

Тематические ссылки:

[gis_data_engine_context_t](#), [gis_data_engine_get_map_list\(\)](#)

gis_data_engine_update()

Обновление области данных движка рендеринга.

Прототип:

```
#include <gis/gisrender.h>

int gis_data_engine_update( gis_data_engine_context_t ctx );
```

Аргументы:

ctx
Указатель на контекст движка рендеринга.

Библиотека:

gisrender

Описание:

Функция обновляет область данных движка рендеринга.

Возвращаемое значение:

<i>EOK</i>	Успешное завершение
<i>EINVAL</i>	Некорректные параметры функции
<i>ENOMEM</i>	Недостаточно памяти
<i>EFAULT</i>	Не удалось обновить область данных

Примеры использования:

```

gis_core_connection_t connection;
gis_data_engine_context_t ctx;
uint32_t width = 100;
uint32_t height = 100;

gis_core_link_init( &connection );
gis_core_link_connect( &connection, 777 );
gis_core_databuffer_attach( &connection );

if ( gis_data_engine_alloc( width, height, GIS_DATA_ENGINE_MODE_SM,
↪ GIS_DATA_ENGINE_BPP_32, &connection, &ctx ) != EOK ) {
    printf( "gis_data_engine_alloc() failed.\n" );
    return (-1);
}

gis_core_request_update_cache( &connection, GIS_CORE_DRIVERS_ALL,
↪ GIS_CORE_UPDATE_CACHE_MODE_SOFT_UPDATE );

gis_core_request_parameters_t map;
gis_core_request_parameters_init( &map );

gis_core_databuffer_data_request( &connection, &map, 1 );

if ( gis_data_engine_update( ctx ) != EOK ) {
    printf( "Failed to update data engine \n" );
    return -1;
}

gis_data_engine_free( &ctx );
gis_core_databuffer_detach();
gis_core_link_destroy( &connection, true );

```

Классификация:

ПК ЦКИ для ЗОСРВ «Нейтрино»

Безопасность использования	
Обработчик прерываний	Нет
Обработчик сигналов	Нет
В потоке	Да

Поддерживается, начиная с ПК ЦКИ редакции 1.0.0

Тематические ссылки:

[gis_data_engine_context_t](#), [gis_data_engine_alloc\(\)](#), [gis_data_engine_free\(\)](#)

gis_data_engine_validity_changed()

Оповещение об изменении состояния движка рендеринга.

Прототип:

```
#include <gis/gisrender.h>

void gis_data_engine_validity_changed( gis_data_engine_context_t ctx, bool isValid );
```

Аргументы:

ctx

Указатель на контекст движка рендеринга

isValid

Флаг состояния движка рендеринга. Доступны следующие варианты: 0 — Движок содержит некорректные данные. 1 — Движок содержит корректные данные.

Библиотека:

gisrender

Описание:

Функция оповещает пользователя об изменении состояния области данных движка рендеринга.

Примеры использования:

```
gis_core_connection_t connection;
gis_data_engine_context_t ctx;
uint32_t width = 100;
uint32_t height = 100;

gis_core_link_init( &connection );
gis_core_link_connect( &connection, 777 );
gis_core_databuffer_attach( &connection );

if ( gis_data_engine_alloc( width, height, GIS_DATA_ENGINE_MODE_SM,
↪ GIS_DATA_ENGINE_BPP_32, &connection, &ctx ) != EOK ) {
    printf( "gis_data_engine_alloc() failed.\n" );
    return 1;
}

gis_data_engine_validity_changed( ctx, false );
```

```
gis_data_engine_validity_changed( ctx, true );

gis_data_engine_free( &ctx );
gis_core_databuffer_detach();
gis_core_link_destroy( &connection, true );
```

Классификация:

ПК ЦКИ для ЗОСРВ «Нейтрино»

Безопасность использования	
<i>Обработчик прерываний</i>	Нет
<i>Обработчик сигналов</i>	Нет
<i>В потоке</i>	Да

Тематические ссылки:

[gis_data_engine_context_t](#), [gis_data_engine_alloc\(\)](#), [gis_data_engine_update\(\)](#)

gis_data_engine_context_t

Контекст движка рендеринга.

Формат:

```
#include <gis/gisrender.h>

typedef void *gis_data_engine_context_t;
```

Описание:

Данный контекст обеспечивает управление параметрами рендеринга ЦКИ библиотекой gisrender.

Классификация:

ПК ЦКИ для ЗОСРВ «Нейтрино»

gis_gui_get_projection_parameters()

Создание диалога установки проекции отображения.

Прототип:

```
#include <gis/gisrender.h>

int gis_gui_get_projection_parameters( gis_map_projection_t *out_proj, void *parent );
```

Аргументы:

out_proj

Указатель на параметры проекции.

parent

Указатель на виджет-предок.

Библиотека:

gisrender

Описание:

Функция создаёт окно установки проекции и заполняет переданные параметры проекции.

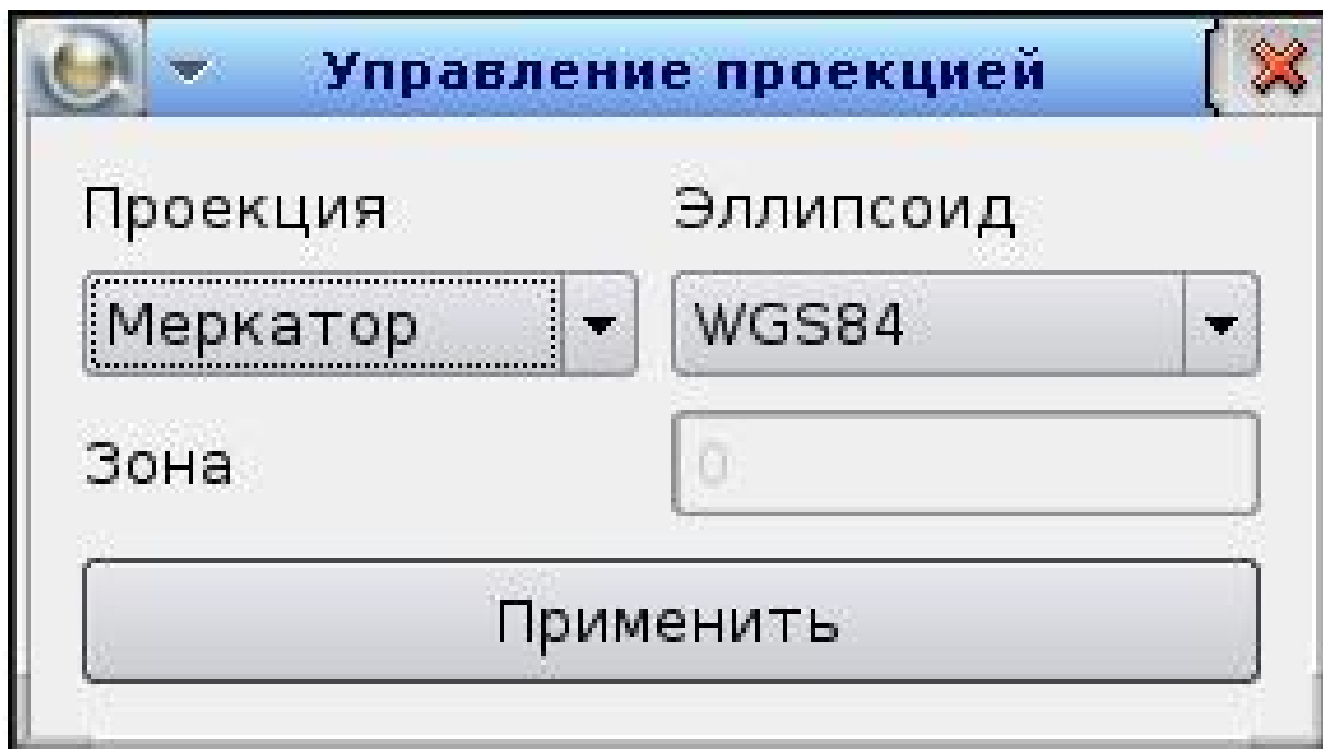


Рисунок 6. Окно установки проекции.

Возвращаемое значение:

EOK

Успешное завершение

EFAULT

Изменения были отменены

Примеры использования:

```
QApplication app( argc, argv );

MapWidget w;
gis_map_projection_t proj_params;
gis_map_projection_init( &proj_params );

gis_gui_get_projection_parameters( &proj_params, &w );
```

Классификация:

Безопасность использования	
Обработчик прерываний	Нет
Обработчик сигналов	Нет
В потоке	Да

Тематические ссылки:

[gis_data_engine_context_t](#), [gis_data_engine_get_map_list\(\)](#), [gis_map_projection_t](#), [gis_data_engine_get_maps_projection_t](#)

gis_gui_get_screen_parameters()

Создание диалога установки параметров экрана.

Прототип:

```
#include <gis/gisrender.h>

int gis_gui_get_screen_parameters( gis_mdp_t mdp, void *parent );
```

Аргументы:

mdp

Контекст `gis_mdp_t`.

parent

Указатель на виджет-предок.

Библиотека:

gisrender

Описание:

Функция создаёт окно установки параметров экрана (размеры и разрешение) и заполняет переданные параметры проекции.

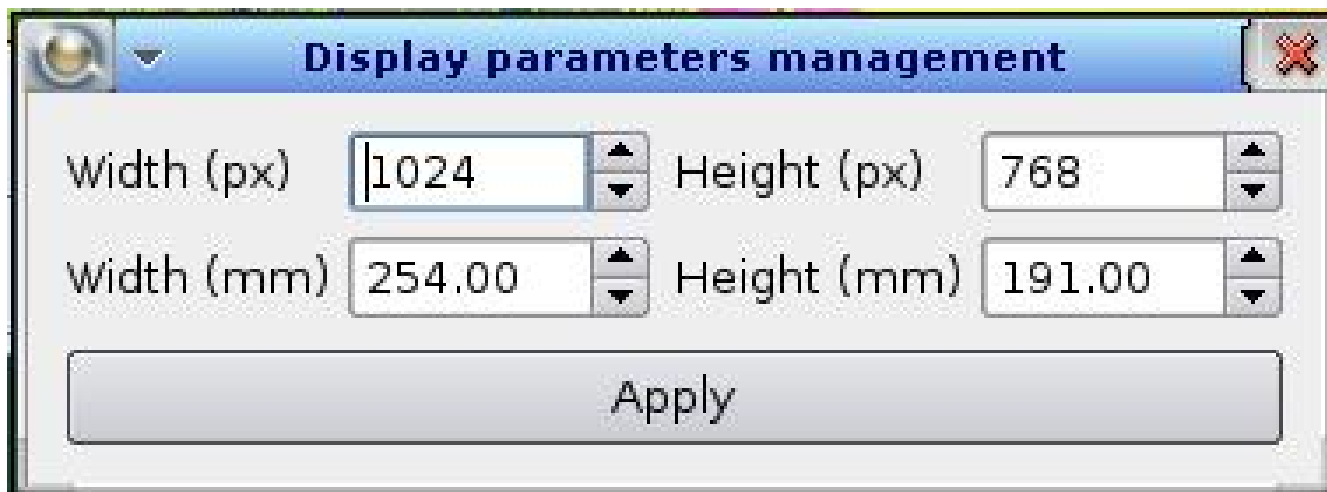


Рисунок 7. Окно установки параметров экрана.

Возвращаемое значение:

EOK

Успешное завершение

EFAULT

Изменения были отменены

Примеры использования:

```
QApplication app( argc, argv );

MapWidget w;
gis_data_engine_context_t ctx;
gis_core_connection_t connection;
uint32_t width = 100;
uint32_t height = 100;
gis_data_engine_alloc( width, height, GIS_DATA_ENGINE_MODE_SM, GIS_DATA_ENGINE_BPP_32,
↪ &connection, &ctx );

gis_mdp_t mdp = gis_data_engine_get_display_parameters( ctx );

gis_gui_get_screen_parameters( mdp, &w );
```

Классификация:

ПК ЦКИ для ЗОСРВ «Нейтрино»

Безопасность использования	
Обработчик прерываний	Нет
Обработчик сигналов	Нет
В потоке	Да

Тематические ссылки:

`gis_mdp_t`, `gis_gui_get_projection_parameters()`

gis_map_ellipsoid_get_full_name()

Получение названия эллипсоида по его индексу.

Прототип:

```
#include <gis/gis_types.h>

const char * gis_map_ellipsoid_get_full_name( gis_map_projection_t *parameters );
```

Аргументы:

parameters

Указатель на расширенный контекст проекции карты.

Библиотека:

gishelper

Описание:

Функция используется для получения названия эллипсоида по расширенному контексту проекции карты. Освобождение памяти, занятое возвращаемой строкой не требуется.

Возвращаемое значение:

pointer

Функция возвращает фиксированную строку с названием эллипсоида

empty

Эллипсоид не найден

Примеры использования:

```
gis_map_projection_t proj;
gis_map_projection_init( &proj );

if ( strcmp (gis_map_ellipsoid_get_full_name( &proj ), "WGS84" ) != 0 )
{
    printf( "Incorrect string: %s\n", gis_map_ellipsoid_get_full_name( &proj ) );
    return 1;
}
```

Классификация:

ПК ЦКИ для ЗОСРВ «Нейтрино»

Безопасность использования	
Обработчик прерываний	Да
Обработчик сигналов	Да
В потоке	Да

Тематические ссылки:

`gis_map_ellipsoid_idx_t`, `gis_map_projection_t`, `gis_map_projection_get_full_name()`, `gis_map_height_system_get_full_name()`

gis_map_height_system_get_full_name()

Получение названия системы высот по её индексу.

Прототип:

```
#include <gis/gis_types.h>

const char * gis_map_height_system_get_full_name( gis_map_height_system_idx_t index );
```

Аргументы:

index

Индекс системы высот `gis_map_height_system_idx_t`.

Библиотека:

gishelper

Описание:

Функция используется для получения названия системы высот по её индексу. Освобождение памяти, занятое возвращаемой строкой не требуется.

Возвращаемое значение:

pointer

Функция возвращает фиксированную строку с названием системы высот

empty

Проекция не найдена

Примеры использования:

```
gis_map_height_system_idx_t idx = GIS_HEIGHT_SYSTEM_BALTIC;

if ( strcmp (gis_map_height_system_get_full_name( idx ), "Zero of the Kronstadt seamark" )
↪ != 0 )
{
    printf( "Incorrect string: %s\n", gis_map_height_system_get_full_name( idx ) );
    return 1;
}
```

Классификация:

ПК ЦКИ для ЗОСРВ «Нейтрино»

Безопасность использования	
Обработчик прерываний	Да
Обработчик сигналов	Да
В потоке	Да

Тематические ссылки:

[gis_map_height_system_idx_t](#), [gis_map_projection_t](#), [gis_map_projection_get_full_name\(\)](#), [gis_map_ellipsoid_get_full_name\(\)](#)

gis_map_projection_get_full_name()

Получение названия проекции по её индексу.

Прототип:

```
#include <gis/gis_types.h>

const char *gis_map_projection_get_full_name( gis_map_projection_t *parameters );
```

Аргументы:

parameters

Указатель на расширенный контекст проекции карты.

Библиотека:

gishelper

Описание:

Функция используется для получения названия проекции по расширенному контексту проекции карты. Освобождение памяти, занятое возвращаемой строкой не требуется.

Возвращаемое значение:

pointer

Функция возвращает фиксированную строку с названием проекции

empty

Проекция не найдена

Примеры использования:

```
gis_map_projection_t proj;
gis_map_projection_init( &proj );

if ( strcmp ( gis_map_projection_get_full_name( &proj ), "Mercator" ) != 0 )
{
    printf( "Incorrect string: %s\n", gis_map_projection_get_full_name( &proj ) );
    return 1;
}
```

Классификация:

ПК ЦКИ для ЗОСРВ «Нейтрино»

Безопасность использования	
Обработчик прерываний	Да
Обработчик сигналов	Да
В потоке	Да

Тематические ссылки:

[gis_map_projection_idx_t](#), [gis_map_projection_t](#), [gis_map_ellipsoid_get_full_name\(\)](#), [gis_map_height_system_get_full_name\(\)](#)

gis_map_projection_has_zone()

Проверка наличия у проекции деления на зоны.

Прототип:

```
#include <gis/gis_types.h>

static inline bool gis_map_projection_has_zone( gis_map_projection_idx_t index );
```

Аргументы:

index

Индекс проекции `gis_map_projection_idx_t`.

Библиотека:

gishelper

Описание:

Функция показывает используется ли в данной проекции параметр 'зона'.

Возвращаемое значение:

true

Параметр используется.

false

Параметр не используется.

Примеры использования:

```
if ( gis_map_projection_has_zone( GIS_PROJECTION_UTM ) )
{
    printf( "Incorrect. UTM has zone\n" );
}
```

Классификация:

Безопасность использования	
Обработчик прерываний	Нет
Обработчик сигналов	Нет
В потоке	Да

Тематические ссылки:

[gis_map_projection_idx_t](#), [gis_map_projection_t](#)

gis_map_projection_init()

Инициализация параметров проекции карты.

Прототип:

```
#include <gis/gis_types.h>

static inline int gis_map_projection_init( gis_map_projection_t *proj );
```

Аргументы:

proj

Указатель на расширенный контекст проекции карты.

Библиотека:

gishelper

Описание:

Функция инициализирует структуру параметров проекции карты (с установкой параметров по умолчанию).

Возвращаемое значение:

EOK

Успешное завершение

EINVAL

Некорректные параметры функции

Примеры использования:

```
gis_map_projection_t    projection;

if ( gis_map_projection_init( &projection ) != EOK )
{
    printf( "Failed to init projection\n" );
    return 1;
}
```

Классификация:

Безопасность использования	
Обработчик прерываний	Нет
Обработчик сигналов	Нет
В потоке	Да

Тематические ссылки:

[gis_map_projection_t](#)

gis_map_projection_is_filled()

Проверка расширенного контекста проекции карты.

Прототип:

```
#include <gis/gis_types.h>

static inline bool gis_map_projection_is_filled( gis_map_projection_t *proj );
```

Аргументы:

proj

Указатель на расширенный контекст проекции карты.

Библиотека:

gishelper

Описание:

Функция проверяет основные поля структуры `gis_map_projection_t`.

Возвращаемое значение:

true

Основные поля заполнены успешно

false

Основные поля не заполнены

Примеры использования:

```
gis_map_projection_t    projection;
gis_map_projection_init( &projection )

if ( !gis_map_projection_is_filled( &projection ) )
{
    printf( "Invalid projection\n" );
    return 1;
}
```

Классификация:

Безопасность использования	
<i>Обработчик прерываний</i>	Нет
<i>Обработчик сигналов</i>	Нет
<i>В потоке</i>	Да

Тематические ссылки:

[gis_map_projection_t](#)

gis_map_projection_zero()

Обнуление параметров проекции.

Прототип:

```
#include <gis/gis_types.h>

static inline int gis_map_projection_zero( gis_map_projection_t *proj );
```

Аргументы:

proj

Указатель на расширенный контекст проекции карты.

Библиотека:

gishelper

Описание:

Функция обнуляет параметры проекции, тем самым делая их некорректными. После успешного исполнения функция [gis_map_projection_is_filled\(\)](#) вернёт false.

Возвращаемое значение:

EOK

Успешное завершение

EINVAL

Некорректные параметры функции

Примеры использования:

```
gis_map_projection_t    projection;

gis_map_projection_zero( &projection );
if ( !gis_map_projection_is_filled( &projection ) )
{
    printf( "Invalid projection\n" );
    return -1;
}
```

Классификация:

ПК ЦКИ для ЗОСРВ «Нейтрино»

Безопасность использования	
Обработчик прерываний	Нет
Обработчик сигналов	Нет
В потоке	Да

Тематические ссылки:

`gis_map_projection_t`, `gis_map_projection_is_filled()`

gis_mdp_get_background_color()

Получение цвета фона карты.

Прототип:

```
#include <gis/gis_mdp.h>

uint32_t gis_mdp_get_background_color( gis_mdp_t mdp_ctx );
```

Аргументы:

mdp_ctx

Контекст `gis_mdp_t`.

Библиотека:

gishelper

Описание:

Функция возвращает цвет фона отображаемой карты. По умолчанию используется 0xFFFFFFFF.

Возвращаемое значение:

background_color

Цвет фона карты.

Примеры использования:

```
gis_mdp_t mdp_ctx;
gis_data_engine_context_t ctx;

gis_core_connection_t connection;
gis_core_link_init( &connection );
gis_core_link_connect( &connection, 777 );
gis_core_databuffer_attach( &connection );

uint32_t width = 100;
uint32_t height = 100;

gis_data_engine_alloc( width, height, GIS_DATA_ENGINE_MODE_SM, GIS_DATA_ENGINE_BPP_32,
↵ &connection, &ctx );
```

```

mdp_ctx = gis_data_engine_get_display_parameters( ctx );

uint32_t background_color = gis_mdp_get_background_color( mdp_ctx );

if ( background_color != 0xFFFFFFFF )
{
    printf( "Incorrect background color: 0x%x", background_color );
    return 1;
}

```

Классификация:

ПК ЦКИ для ЗОСРВ «Нейтрино»

Безопасность использования	
Обработчик прерываний	Нет
Обработчик сигналов	Нет
В потоке	Да

Поддерживается, начиная с ПК ЦКИ редакции 1.1.0

Тематические ссылки:

[gis_mdp_t](#), [gis_mdp_set_background_color\(\)](#)

gis_mdp_get_brightness_contrast()

Получение значений яркости и контрастности отображения.

Прототип:

```
#include <gis/gis_mdp.h>

int gis_mdp_get_brightness_contrast( gis_mdp_t mdp_ctx, float *brightness, float *contrast
↵ );
```

Аргументы:

mdp_ctx

Контекст `gis_mdp_t`.

brightness

Указатель на значение яркости.

contrast

Указатель на значение контрастности.

Библиотека:

gishelper

Описание:

Функция получает значения яркости и контрастности отображения карты.

Возвращаемое значение:

EOK

Успешное завершение

EINVAL

Некорректный контекст `gis_mdp_t`

Примеры использования:

```

gis_mdp_t mdp_ctx;
gis_data_engine_context_t ctx;
gis_core_connection_t connection;
gis_core_link_init( &connection );
gis_core_link_connect( &connection, 777 );
gis_core_databuffer_attach( &connection );

uint32_t width = 100;
uint32_t height = 100;

gis_data_engine_alloc( width, height, GIS_DATA_ENGINE_MODE_SM, GIS_DATA_ENGINE_BPP_32,
↳ &connection, &ctx );

mdp_ctx = gis_data_engine_get_display_parameters( ctx );

if ( gis_mdp_set_brightness_contrast( mdp_ctx, 0.5, 0.5 ) != EOK ) {
    printf( "Failed to set brightness and contrast parameters\n" );
    return 1;
}

float test_brightness, test_contrast;
if ( gis_mdp_get_brightness_contrast( mdp_ctx, &test_brightness, &test_contrast ) != EOK
↳ ) {
    printf( "Failed to get brightness and contrast parameters\n" );
    return 1;
}

if ( fabs( test_brightness - 0.5 ) < 0.01 &&
    fabs( test_contrast - 0.5 ) < 0.01 ) {
    printf( "Correct\n" );
}
else
{
    printf( "Incorrect brightness and contrast parameters\n" );
    return 1;
}

```

Классификация:

ПК ЦКИ для ЗОСРВ «Нейтрино»

Безопасность использования	
Обработчик прерываний	Нет
Обработчик сигналов	Нет
В потоке	Да

Поддерживается, начиная с ПК ЦКИ редакции 1.2

Тематические ссылки:

`gis_mdp_t`, `gis_mdp_set_brightness_contrast()`

gis_mdp_get_center_point_deg()

Получение градусных координат центра отображения карты.

Прототип:

```
#include <gis/gis_mdp.h>

int gis_mdp_get_center_point_deg( gis_mdp_t mdp_ctx, double_point_t *point );
```

Аргументы:

mdp_ctx

Контекст `gis_mdp_t`.

point

Указатель на центральную точку (долгота, широта).

Библиотека:

gishelper

Описание:

Функция возвращает градусные координаты центра отображения карты.

Возвращаемое значение:

EOK

Успешное завершение.

EINVAL

Некорректные параметры функции.

Примеры использования:

```
gis_mdp_t mdp_ctx;
gis_data_engine_context_t ctx;
gis_core_connection_t connection;
gis_core_link_init( &connection );
gis_core_link_connect( &connection, 777 );
gis_core_databuffer_attach( &connection );

uint32_t width = 600;
uint32_t height = 600;
```

```

    gis_data_engine_alloc( width, height, GIS_DATA_ENGINE_MODE_SM, GIS_DATA_ENGINE_BPP_32,
↪ &connection, &ctx );

    mdp_ctx = gis_data_engine_get_display_parameters( ctx );

    double_point_t point = { 40, 45 };

    if ( gis_mdp_set_center_point( mdp_ctx, point ) != EOK ) {
        printf( "Failed to set center point\n" );
        return 1;
    }

    if ( gis_mdp_get_center_point_deg( mdp_ctx, &point ) != EOK ) {
        printf( "Failed to get center point\n" );
        return 1;
    }

    if ( point.x != 40 && point.y != 45 ) {
        printf( "Incorrect center point\n" );
        return 1;
    }

```

Классификация:

ПК ЦКИ для ЗОСРВ «Нейтрино»

Безопасность использования	
Обработчик прерываний	Нет
Обработчик сигналов	Нет
В потоке	Да

Поддерживается, начиная с ПК ЦКИ редакции 1.0.0

Тематические ссылки:

[gis_mdp_t](#), [gis_mdp_set_center_point\(\)](#) [gis_mdp_set_display_resolution\(\)](#)

gis_mdp_get_display_resolution()

Получение установленного разрешения экрана в пикселях.

Прототип:

```
#include <gis/gis_mdp.h>

void gis_mdp_get_display_resolution( gis_mdp_t mdp_ctx, int* width_px, int* height_px );
```

Аргументы:

mdp_ctx

Контекст `gis_mdp_t`

width_px

Указатель на ширину экрана.

height_px

Указатель на высоту экрана.

Библиотека:

gishelper

Описание:

Функция возвращает установленное разрешение экрана для текущего отображения карты.

Примеры использования:

```
gis_mdp_t mdp_ctx;
gis_data_engine_context_t ctx;
gis_core_connection_t connection;
gis_core_link_init( &connection );
gis_core_link_connect( &connection, 777 );
gis_core_databuffer_attach( &connection );

uint32_t width = 600;
uint32_t height = 600;

gis_data_engine_alloc( width, height, GIS_DATA_ENGINE_MODE_SM, GIS_DATA_ENGINE_BPP_32,
↵ &connection, &ctx );

mdp_ctx = gis_data_engine_get_display_parameters( ctx );
```

```

int width_res = 0;
int height_res = 0;

if ( gis_mdp_set_display_resolution( mdp_ctx, 800, 600 ) != EOK ) {
    printf( "Failed to set resolution\n" );
    return 1;
}

gis_mdp_get_display_resolution( mdp_ctx, &width_res, &height_res );

if ( width_res != 800 && height_res != 600 )
{
    printf( "Incorrect resolution: %d %d", width_res, height_res );
    return 1;
}

```

Классификация:

ПК ЦКИ для ЗОСРВ «Нейтрино»

Безопасность использования	
Обработчик прерываний	Нет
Обработчик сигналов	Нет
В потоке	Да

Тематические ссылки:

[gis_mdp_t](#), [gis_mdp_set_display_resolution\(\)](#)

gis_mdp_get_display_size()

Получение установленного размера экрана в миллиметрах.

Прототип:

```
#include <gis/gis_mdp.h>

void gis_mdp_get_display_size( gis_mdp_t mdp_ctx, double* width_mm, double* height_mm );
```

Аргументы:

mdp_ctx

Контекст `gis_mdp_t`.

width_mm

Указатель на ширину экрана.

height_mm

Указатель на высоту экрана.

Библиотека:

gishelper

Описание:

Функция возвращает размер экрана для текущего отображения карты.

Примеры использования:

```
gis_mdp_t mdp_ctx;
gis_data_engine_context_t ctx;
gis_core_connection_t connection;
gis_core_link_init( &connection );
gis_core_link_connect( &connection, 777 );
gis_core_databuffer_attach( &connection );

uint32_t width = 600;
uint32_t height = 600;

gis_data_engine_alloc( width, height, GIS_DATA_ENGINE_MODE_SM, GIS_DATA_ENGINE_BPP_32,
↵ &connection, &ctx );

mdp_ctx = gis_data_engine_get_display_parameters( ctx );
```

```

double width_res = 0;
double height_res = 0;

if ( gis_mdp_set_display_size( mdp_ctx, 510, 290 ) != EOK ) {
    printf( "Failed to set display size\n" );
    return 1;
}

gis_mdp_get_display_size( mdp_ctx, &width_res, &height_res );

if ( width_res != 510 && height_res != 290 )
{
    printf( "Incorrect display size: %f %f\n", width_res, height_res );
    return 1;
}

```

Классификация:

ПК ЦКИ для ЗОСРВ «Нейтрино»

Безопасность использования	
Обработчик прерываний	Нет
Обработчик сигналов	Нет
В потоке	Да

Тематические ссылки:

[gis_mdp_t](#), [gis_mdp_set_display_size\(\)](#)

gis_mdp_get_map_antialiasing_level()

Получение текущего уровня сглаживания карты.

Прототип:

```
#include <gis/gis_mdp.h>

int gis_mdp_get_map_antialiasing_level( gis_mdp_t mdp_ctx );
```

Аргументы:

mdp_ctx

Контекст `gis_mdp_t`.

Библиотека:

gishelper

Описание:

Функция получает уровень сглаживания отображения карты.

Возвращаемое значение:

level

Уровень сглаживания. Возможны следующие варианты: 0 — Без сглаживания. 1 — Сглаживание включено.

-(*EINVAL*)

Некорректный контекст `gis_mdp_t`.

Примеры использования:

```
gis_mdp_t mdp_ctx;
gis_data_engine_context_t ctx;
gis_core_connection_t connection;
gis_core_link_init( &connection );
gis_core_link_connect( &connection, 777 );
gis_core_databuffer_attach( &connection );

uint32_t width = 600;
uint32_t height = 600;
```

```

gis_data_engine_alloc( width, height, GIS_DATA_ENGINE_MODE_SM, GIS_DATA_ENGINE_BPP_32,
↪ &connection, &ctx );

mdp_ctx = gis_data_engine_get_display_parameters( ctx );

if ( gis_mdp_set_map_antialiasing_level( mdp_ctx, 1 ) != EOK ) {
    printf( "Failed to set antialiasing level\n" );
    return 1;
}

if ( gis_mdp_get_map_antialiasing_level( mdp_ctx ) != 1 ) {
    printf( "Incorrect antialiasing level\n" );
    return 1;
}

```

Классификация:

ПК ЦКИ для ЗОСРВ «Нейтрино»

Безопасность использования	
Обработчик прерываний	Нет
Обработчик сигналов	Нет
В потоке	Да

Тематические ссылки:

`gis_mdp_t`, `gis_mdp_set_scaling_mode()`

gis_mdp_get_phys_scale()

Получение текущего масштаба отображения карты.

Прототип:

```
#include <gis/gis_mdp.h>

uint32_t gis_mdp_get_phys_scale( gis_mdp_t mdp_ctx );
```

Аргументы:

mdp_ctx

Контекст `gis_mdp_t`.

Библиотека:

gishelper

Описание:

Функция возвращает масштаб текущего отображения карты.

Возвращаемое значение:

uint32_t

Успешное завершение.

0

Некорректный контекст.

Примеры использования:

```
gis_mdp_t mdp_ctx;
gis_data_engine_context_t ctx;
gis_core_connection_t connection;
gis_core_link_init( &connection );
gis_core_link_connect( &connection, 777 );
gis_core_databuffer_attach( &connection );

uint32_t width = 600;
uint32_t height = 600;

gis_data_engine_alloc( width, height, GIS_DATA_ENGINE_MODE_SM, GIS_DATA_ENGINE_BPP_32,
↪ &connection, &ctx );
```

```

mdp_ctx = gis_data_engine_get_display_parameters( ctx );

if ( gis_mdp_set_phys_scale( mdp_ctx, 100000 ) != EOK ) {
    printf( "Failed to set scale\n" );
    return 1;
}

uint32_t scale = gis_mdp_get_phys_scale( mdp_ctx );

if ( scale != 100000 ) {
    printf( "Incorrect scale: %d\n", scale );
    return 1;
}

```

Классификация:

ПК ЦКИ для ЗОСРВ «Нейтрино»

Безопасность использования	
Обработчик прерываний	Нет
Обработчик сигналов	Нет
В потоке	Да

Тематические ссылки:

[gis_mdp_t](#), [gis_mdp_set_projection\(\)](#), [gis_mdp_set_phys_scale\(\)](#)

gis_mdp_get_projection()

Получение текущей проекции отображения карты.

Прототип:

```
#include <gis/gis_mdp.h>

gis_map_projection_t *gis_mdp_get_projection( gis_mdp_t mdp_ctx );
```

Аргументы:

mdp_ctx

Контекст `gis_mdp_t`.

Библиотека:

gishelper

Описание:

Функция возвращает проекцию текущего отображения карты.

Возвращаемое значение:

*gis_map_projection_t**

Указатель на структуру параметров проекции, тип `gis_map_projection_t`

NULL

Некорректный контекст.

Примеры использования:

```
gis_mdp_t mdp_ctx;
gis_data_engine_context_t ctx;
gis_core_connection_t connection;
gis_core_link_init( &connection );
gis_core_link_connect( &connection, 777 );
gis_core_databuffer_attach( &connection );

uint32_t width = 600;
uint32_t height = 600;

gis_data_engine_alloc( width, height, GIS_DATA_ENGINE_MODE_SM, GIS_DATA_ENGINE_BPP_32,
↪ &connection, &ctx );
```

```

mdp_ctx = gis_data_engine_get_display_parameters( ctx );

gis_map_projection_t projection;
gis_map_projection_init( &projection );
projection.projection_idx = GIS_PROJECTION_GAUSS_KRUGER_CONFORMAL;
projection.ellipsoid_idx  = GIS_ELLIPSOID_KRASSOVSKY_1942;
projection.zone           = 7;

if ( gis_mdp_set_projection( mdp_ctx, &projection ) != EOK ) {
    printf( "Failed to set projection\n" );
    return 1;
}

gis_map_projection_t *proj = gis_mdp_get_projection( mdp_ctx );
if ( proj->projection_idx != GIS_PROJECTION_GAUSS_KRUGER_CONFORMAL &&
    proj->ellipsoid_idx  != GIS_ELLIPSOID_KRASSOVSKY_1942 &&
    proj->zone           != 7 )
{
    printf( "Incorrect projection parameters\n" );
    return 1;
}

```

Классификация:

ПК ЦКИ для ЗОСРВ «Нейтрино»

Безопасность использования	
Обработчик прерываний	Нет
Обработчик сигналов	Нет
В потоке	Да

Поддерживается, начиная с ПК ЦКИ редакции 1.0.0

Тематические ссылки:

[gis_mdp_t](#), [gis_mdp_set_projection\(\)](#)

gis_mdp_get_raster_color_mode()

Получение типа заливки для рисования раstra.

Прототип:

```
#include <gis/gis_mdp.h>

bool gis_mdp_get_raster_color_mode( gis_mdp_t mdp_ctx );
```

Аргументы:

mdp_ctx

Контекст `gis_mdp_t`.

Библиотека:

gishelper

Описание:

Функция получает тип заливки для рисования растровых файлов.

Возвращаемое значение:

false

Градиентная заливка. Используется по умолчанию.

true

Заливка дискретными цветами из палитры.

Примеры использования:

```
gis_mdp_t mdp_ctx;
gis_data_engine_context_t ctx;
gis_core_connection_t connection;
gis_core_link_init( &connection );
gis_core_link_connect( &connection, 777 );
gis_core_databuffer_attach( &connection );

uint32_t width = 600;
uint32_t height = 600;

gis_data_engine_alloc( width, height, GIS_DATA_ENGINE_MODE_SM, GIS_DATA_ENGINE_BPP_32,
↪ &connection, &ctx );
```

```

mdp_ctx = gis_data_engine_get_display_parameters( ctx );

gis_mdp_set_raster_color_mode( mdp_ctx, false );

if ( gis_mdp_get_raster_color_mode( mdp_ctx ) )
{
    printf( "Failed to get raster color mode\n" );
    return 1;
}

```

Классификация:

ПК ЦКИ для ЗОСРВ «Нейтрино»

Безопасность использования	
Обработчик прерываний	Нет
Обработчик сигналов	Нет
В потоке	Да

Поддерживается, начиная с ПК ЦКИ редакции 1.2

Тематические ссылки:

gis_mdp_t, gis_mdp_set_raster_color_mode() gis_mdp_set_raster_height_mode() gis_mdp_get_raster_height_mode()

gis_mdp_get_raster_height_mode()

Получение режима выбора границ высот для растровых файлов.

Прототип:

```
#include <gis/gis_mdp.h>

bool gis_mdp_get_raster_height_mode( gis_mdp_t mdp_ctx );
```

Аргументы:

mdp_ctx

Контекст `gis_mdp_t`.

Библиотека:

gishelper

Описание:

Функция возвращает режим выбора границ высот для отображения растровых файлов.

Возвращаемое значение:

bool

Флаг, задающий режим выбора высот. Режимы работы: false — ручная установка диапазона высот. true — автоматическая настройка диапазона высот по доступным картам. Используется по умолчанию.

Примеры использования:

```
gis_mdp_t mdp_ctx;
gis_data_engine_context_t ctx;
gis_core_connection_t connection;
gis_core_link_init( &connection );
gis_core_link_connect( &connection, 777 );
gis_core_databuffer_attach( &connection );

uint32_t width = 600;
uint32_t height = 600;

gis_data_engine_alloc( width, height, GIS_DATA_ENGINE_MODE_SM, GIS_DATA_ENGINE_BPP_32,
↪ &connection, &ctx );
```

```

mdp_ctx = gis_data_engine_get_display_parameters( ctx );

gis_mdp_set_raster_height_mode( mdp_ctx, false );

if ( gis_mdp_get_raster_height_mode( mdp_ctx ) != false )
{
    printf( "Incorrect height mode\n" );
    return 1;
}

```

Классификация:

ПК ЦКИ для ЗОСРВ «Нейтрино»

Безопасность использования	
Обработчик прерываний	Нет
Обработчик сигналов	Нет
В потоке	Да

Тематические ссылки:

[gis_mdp_t](#), [gis_mdp_set_raster_height_mode\(\)](#)

gis_mdp_get_raster_invalid_height_color()

Получение цвета отображения высот вне заданного диапазона.

Прототип:

```
#include <gis/gis_mdp.h>

uint32_t gis_mdp_get_raster_invalid_height_color( gis_mdp_t mdp_ctx );
```

Аргументы:

mdp_ctx

Контекст `gis_mdp_t`.

Библиотека:

gishelper

Описание:

Функция получает цвет отображения высот вне заданного диапазона для растровых файлов.

Возвращаемое значение:

uint32_t

Успешное завершение.

0

Некорректный контекст.

Примеры использования:

```
gis_mdp_t mdp_ctx;
gis_data_engine_context_t ctx;
gis_core_connection_t connection;
gis_core_link_init( &connection );
gis_core_link_connect( &connection, 777 );
gis_core_databuffer_attach( &connection );

uint32_t width = 600;
uint32_t height = 600;

gis_data_engine_alloc( width, height, GIS_DATA_ENGINE_MODE_SM, GIS_DATA_ENGINE_BPP_32,
↪ &connection, &ctx );
```

```

mdp_ctx = gis_data_engine_get_display_parameters( ctx );

gis_mdp_set_raster_invalid_height_color( mdp_ctx, 0xFF00EE );

if ( gis_mdp_get_raster_invalid_height_color( mdp_ctx ) != 0xFF00EE )
{
    printf( "Failed to get invalid color value\n" );
    return 1;
}

```

Классификация:

ПК ЦКИ для ЗОСРВ «Нейтрино»

Безопасность использования	
Обработчик прерываний	Нет
Обработчик сигналов	Нет
В потоке	Да

Тематические ссылки:

[gis_mdp_t](#), [gis_mdp_set_raster_invalid_height_color\(\)](#), [gis_mdp_set_raster_color_mode\(\)](#), [gis_mdp_get_raster_lower_h](#)

Поддерживается, начиная с ПК ЦКИ редакции 1.2

gis_mdp_get_raster_lower_height_limit()

Получение нижнего предела высот отображения для растровых файлов.

Прототип:

```
#include <gis/gis_mdp.h>

double gis_mdp_get_raster_lower_height_limit( gis_mdp_t mdp_ctx );
```

Аргументы:

mdp_ctx

Контекст `gis_mdp_t`.

Библиотека:

gishelper

Описание:

Функция возвращает нижний предел высот отображения для растровых файлов. Высоты меньше этого значения будут отображаться цветом `invalid_height_color`, заданным в `gis_mdp_set_raster_palette()`

Возвращаемое значение:

double

Нижняя граница диапазона высот.

9999

Некорректный указатель на контекст.

Примеры использования:

```
gis_mdp_t mdp_ctx;
gis_data_engine_context_t ctx;
gis_core_connection_t connection;
gis_core_link_init( &connection );
gis_core_link_connect( &connection, 777 );
gis_core_databuffer_attach( &connection );

uint32_t width = 600;
uint32_t height = 600;
```

```

gis_data_engine_alloc( width, height, GIS_DATA_ENGINE_MODE_SM, GIS_DATA_ENGINE_BPP_32,
↪ &connection, &ctx );

mdp_ctx = gis_data_engine_get_display_parameters( ctx );

gis_mdp_set_raster_height_limits( mdp_ctx, 40, 200 );

double upper_limit = gis_mdp_get_raster_upper_height_limit( mdp_ctx );
double lower_limit = gis_mdp_get_raster_lower_height_limit( mdp_ctx );

if ( upper_limit != 200 && lower_limit != 40 )
{
    printf( "Incorrect height limits\n" );
    return 1;
}

```

Классификация:

ПК ЦКИ для ЗОСРВ «Нейтрино»

Безопасность использования	
Обработчик прерываний	Нет
Обработчик сигналов	Нет
В потоке	Да

Тематические ссылки:

[gis_mdp_t](#), [gis_mdp_get_raster_upper_height_limit\(\)](#)

gis_mdp_get_raster_palette()

Получение палитры для рисования раstra.

Прототип:

```
#include <gis/gis_mdp.h>

uint32_t * gis_mdp_get_raster_palette( gis_mdp_t mdp_ctx, uint32_t *color_number );
```

Аргументы:

mdp_ctx

Контекст `gis_mdp_t`.

color_number

Указатель на количество цветов в массиве цветовой палитры.

Библиотека:

gishelper

Описание:

Функция устанавливает палитру рисования растровых карт.

Возвращаемое значение:

NULL

Некорректный контекст.

!NULL

Указатель на выделенную память, содержащую палитру в виде массива значений типа `uint32_t`.
Количество элементов массива определяется аргументом `color_number`.



По окончании пользования значениями из полученной палитры, необходимо освободить память, выделенную под палитру путем вызова функции `free()`.

Примеры использования:

```

gis_mdp_t mdp_ctx;
gis_data_engine_context_t ctx;
gis_core_connection_t connection;
gis_core_link_init( &connection );
gis_core_link_connect( &connection, 777 );
gis_core_databuffer_attach( &connection );

uint32_t width = 600;
uint32_t height = 600;

gis_data_engine_alloc( width, height, GIS_DATA_ENGINE_MODE_SM, GIS_DATA_ENGINE_BPP_32,
↳ &connection, &ctx );

mdp_ctx = gis_data_engine_get_display_parameters( ctx );

vector<uint32_t> colorMap;
colorMap.clear();
colorMap.push_back( 0x00004030 );
colorMap.push_back( 0x00186424 );
colorMap.push_back( 0x0048AC0C );
colorMap.push_back( 0x00ACC800 );
colorMap.push_back( 0x00D0E000 );
colorMap.push_back( 0x00F4F800 );
colorMap.push_back( 0x00E4C400 );
colorMap.push_back( 0x00D29A00 );
colorMap.push_back( 0x00C07000 );
colorMap.push_back( 0x00AE4600 );
colorMap.push_back( 0x00A22A00 );

if ( gis_mdp_set_raster_palette( mdp_ctx, colorMap.data(), colorMap.size() ) != EOK )
{
    printf( "Failed to set palette\n" );
    return 1;
}

uint32_t *get_palette;
uint32_t number_of_colors;
if ( ( get_palette = gis_mdp_get_raster_palette( mdp_ctx, &number_of_colors ) ) != NULL
↳ )
{
    if ( *get_palette != 0x00004030 ||
        *(get_palette + 1) != 0x00186424 )
    {
        printf( "Failed to get palette\n" );
        return 1;
    }
}

free( get_palette );

```

Классификация:

ПК ЦКИ для ЗОСРВ «Нейтрино»

Безопасность использования	
Обработчик прерываний	Нет
Обработчик сигналов	Нет
В потоке	Да

Поддерживается, начиная с ПК ЦКИ редакции 1.2

Тематические ссылки:

`gis_mdp_t`, `gis_mdp_set_raster_palette()`, `gis_mdp_set_raster_color_mode()`

gis_mdp_get_raster_upper_height_limit()

Получение верхнего предела высот отображения для растровых файлов.

Прототип:

```
#include <gis/gis_mdp.h>

double gis_mdp_get_raster_upper_height_limit( gis_mdp_t mdp_ctx );
```

Аргументы:

mdp_ctx

Контекст `gis_mdp_t`.

Библиотека:

gishelper

Описание:

Функция возвращает верхний предел высот отображения для растровых файлов. Высоты больше этого значения будут отображаться цветом `invalid_height_color`, заданным в `gis_mdp_set_raster_palette()`

Возвращаемое значение:

double

Верхняя граница диапазона высот.

-9999

Некорректный указатель на контекст.

Примеры использования:

```
gis_mdp_t mdp_ctx;
gis_data_engine_context_t ctx;
gis_core_connection_t connection;
gis_core_link_init( &connection );
gis_core_link_connect( &connection, 777 );
gis_core_databuffer_attach( &connection );

uint32_t width = 600;
uint32_t height = 600;
```

```

gis_data_engine_alloc( width, height, GIS_DATA_ENGINE_MODE_SM, GIS_DATA_ENGINE_BPP_32,
↪ &connection, &ctx );

mdp_ctx = gis_data_engine_get_display_parameters( ctx );

gis_mdp_set_raster_height_limits( mdp_ctx, 40, 200 );

double upper_limit = gis_mdp_get_raster_upper_height_limit( mdp_ctx );
double lower_limit = gis_mdp_get_raster_lower_height_limit( mdp_ctx );

if ( upper_limit != 200 && lower_limit != 40 )
{
    printf( "Incorrect height limits\n" );
    return 1;
}

```

Классификация:

ПК ЦКИ для ЗОСРВ «Нейтрино»

Безопасность использования	
Обработчик прерываний	Нет
Обработчик сигналов	Нет
В потоке	Да

Тематические ссылки:

[gis_mdp_t](#), [gis_mdp_get_raster_lower_height_limit\(\)](#)

gis_mdp_get_scaling_mode()

Получение режима масштабирования для текущего отображения карты.

Прототип:

```
#include <gis/gis_mdp.h>

bool gis_mdp_get_scaling_mode( gis_mdp_t mdp_ctx );
```

Аргументы:

mdp_ctx

Контекст `gis_mdp_t`.

Библиотека:

gishelper

Описание:

Функция возвращает режим масштабирования для текущего отображения карты.

Возвращаемое значение:

constant_scale_mode

Режим масштабирования. true - картографический режим. В этом режиме толщина линий (в пикселях) постоянна. false - чертежный режим.

Примеры использования:

```
gis_mdp_t mdp_ctx;
gis_data_engine_context_t ctx;
gis_core_connection_t connection;
gis_core_link_init( &connection );
gis_core_link_connect( &connection, 777 );
gis_core_databuffer_attach( &connection );

uint32_t width = 600;
uint32_t height = 600;

gis_data_engine_alloc( width, height, GIS_DATA_ENGINE_MODE_SM, GIS_DATA_ENGINE_BPP_32,
↵ &connection, &ctx );
```

```

mdp_ctx = gis_data_engine_get_display_parameters( ctx );

if ( gis_mdp_set_scaling_mode( mdp_ctx, true ) != EOK ) {
    printf( "Failed to set mode\n" );
    return 1;
}

if ( gis_mdp_get_scaling_mode( mdp_ctx ) != true ) {
    printf( "Invalid mode\n" );
    return 1;
}

```

Классификация:

ПК ЦКИ для ЗОСРВ «Нейтрино»

Безопасность использования	
Обработчик прерываний	Нет
Обработчик сигналов	Нет
В потоке	Да

Тематические ссылки:

[gis_mdp_t](#), [gis_mdp_set_scaling_mode\(\)](#)

gis_mdp_set_background_color()

Установка цвета фона карты.

Прототип:

```
#include <gis/gis_mdp.h>

int gis_mdp_set_background_color( gis_mdp_t mdp_ctx, uint32_t color );
```

Аргументы:

mdp_ctx

Контекст `gis_mdp_t`.

color

Цвет фона в формате ARGB.

Библиотека:

gishelper

Описание:

Функция устанавливает цвет фона отображаемой карты. По умолчанию используется 0xFFFFFFFF.

Возвращаемое значение:

EOK

Успешное завершение.

EINVAL

Некорректные параметры функции.

Примеры использования:

```
gis_mdp_t mdp_ctx;
gis_data_engine_context_t ctx;
gis_core_connection_t connection;
gis_core_link_init( &connection );
gis_core_link_connect( &connection, 777 );
gis_core_databuffer_attach( &connection );

uint32_t width = 100;
uint32_t height = 100;
```

```

    gis_data_engine_alloc( width, height, GIS_DATA_ENGINE_MODE_SM, GIS_DATA_ENGINE_BPP_32,
↪ &connection, &ctx );

    mdp_ctx = gis_data_engine_get_display_parameters( ctx );

    if ( gis_mdp_set_background_color( mdp_ctx, 0xffffffff ) != EOK ) {
        printf( "Failed to set background color\n" );
        return 1;
    }

    uint32_t background_color = gis_mdp_get_background_color( mdp_ctx );
    if ( background_color != 0xffffffff )
    {
        printf( "Incorrect background color: 0x%x\n", background_color );
        return 1;
    }

```

Классификация:

ПК ЦКИ для ЗОСРВ «Нейтрино»

Безопасность использования	
Обработчик прерываний	Нет
Обработчик сигналов	Нет
В потоке	Да

Тематические ссылки:

[gis_mdp_t](#), [gis_mdp_set_display_resolution\(\)](#)

gis_mdp_set_brightness_contrast()

Установка яркости и контрастности отображения.

Прототип:

```
#include <gis/gis_mdp.h>

int gis_mdp_set_brightness_contrast( gis_mdp_t mdp_ctx, float brightness, float contrast );
```

Аргументы:

mdp_ctx

Контекст `gis_mdp_t`.

brightness

Новая яркость. Задаётся в диапазоне от -1 до 1 включительно.

contrast

Новая контрастность. Задаётся в диапазоне от -1 до 1 включительно.

Библиотека:

gishelper

Описание:

Функция устанавливает яркость и контрастность отображения карты.

Возвращаемое значение:

EOK

Успешное завершение

EINVAL

Некорректные параметры функции

Примеры использования:

```

gis_mdp_t mdp_ctx;
gis_data_engine_context_t ctx;
gis_core_connection_t connection;
gis_core_link_init( &connection );
gis_core_link_connect( &connection, 777 );
gis_core_databuffer_attach( &connection );

uint32_t width = 100;
uint32_t height = 100;

gis_data_engine_alloc( width, height, GIS_DATA_ENGINE_MODE_SM, GIS_DATA_ENGINE_BPP_32,
↳ &connection, &ctx );

mdp_ctx = gis_data_engine_get_display_parameters( ctx );

if ( gis_mdp_set_brightness_contrast( mdp_ctx, 0.5, 0.5 ) != EOK ) {
    printf( "Failed to set brightness and contrast parameters\n" );
    return 1;
}

float test_brightness, test_contrast;
if ( gis_mdp_get_brightness_contrast( mdp_ctx, &test_brightness, &test_contrast ) != EOK
↳ ) {
    printf( "Failed to get brightness and contrast parameters\n" );
    return 1;
}

if ( fabs( test_brightness - 0.5 ) < 0.01 &&
    fabs( test_contrast - 0.5 ) < 0.01 ) {
    printf( "Correct\n" );
}
else
{
    printf( "Incorrect brightness and contrast parameters\n" );
    return 1;
}

```

Классификация:

ПК ЦКИ для ЗОСРВ «Нейтрино»

Безопасность использования	
Обработчик прерываний	Нет
Обработчик сигналов	Нет
В потоке	Да

Тематические ссылки:

```
gis_mdp_t, gis_mdp_set_background_color() gis_mdp_get_brightness_contrast()
```

gis_mdp_set_center_point()

Установка центра отображения карты.

Прототип:

```
#include <gis/gis_mdp.h>

int gis_mdp_set_center_point( gis_mdp_t mdp_ctx, double_point_t point );
```

Аргументы:

mdp_ctx

Контекст `gis_mdp_t`.

point

Центральная точка (долгота, широта).

Библиотека:

gishelper

Описание:

Функция устанавливает географическую центральную точку отображения карты.

Возвращаемое значение:

EOK

Успешное завершение.

EINVAL

Некорректные параметры функции.

Примеры использования:

```
gis_mdp_t mdp_ctx;
gis_data_engine_context_t ctx;
gis_core_connection_t connection;
gis_core_link_init( &connection );
gis_core_link_connect( &connection, 777 );
gis_core_databuffer_attach( &connection );

uint32_t width = 600;
uint32_t height = 600;
```

```

    gis_data_engine_alloc( width, height, GIS_DATA_ENGINE_MODE_SM, GIS_DATA_ENGINE_BPP_32,
↪ &connection, &ctx );

    mdp_ctx = gis_data_engine_get_display_parameters( ctx );

    double_point_t point = { 40, 45 };

    if ( gis_mdp_set_center_point( mdp_ctx, point ) != EOK ) {
        printf( "Failed to set center point\n" );
        return 1;
    }

    if ( gis_mdp_get_center_point_deg( mdp_ctx, &point ) != EOK ) {
        printf( "Failed to get center point\n" );
        return 1;
    }

    if ( point.x != 40 && point.y != 45 ) {
        printf( "Incorrect center point\n" );
        return 1;
    }

```

Классификация:

ПК ЦКИ для ЗОСРВ «Нейтрино»

Безопасность использования	
Обработчик прерываний	Нет
Обработчик сигналов	Нет
В потоке	Да

Тематические ссылки:

[gis_mdp_t](#), [gis_mdp_set_center_point_ptr\(\)](#)

gis_mdp_set_center_point_ptr()

Установка указателя на центр отображения карты.

Прототип:

```
#include <gis/gis_mdp.h>

int gis_mdp_set_center_point_ptr( gis_mdp_t mdp_ctx, double_point_t *point );
```

Аргументы:

mdp_ctx

Контекст `gis_mdp_t`.

point

Указатель на центральную точку (долгота, широта).

Библиотека:

gishelper

Описание:

Функция устанавливает указатель на центральную точку отображения карты.

Возвращаемое значение:

EOK

Успешное завершение.

EINVAL

Некорректные параметры функции.

Примеры использования:

```
gis_mdp_t mdp_ctx;
gis_data_engine_context_t ctx;
gis_core_connection_t connection;
gis_core_link_init( &connection );
gis_core_link_connect( &connection, 777 );
gis_core_databuffer_attach( &connection );

uint32_t width = 600;
uint32_t height = 600;
```

```

    gis_data_engine_alloc( width, height, GIS_DATA_ENGINE_MODE_SM, GIS_DATA_ENGINE_BPP_32,
↪ &connection, &ctx );

    mdp_ctx = gis_data_engine_get_display_parameters( ctx );

    double_point_t new_center = { 40, 45 };
    double_point_t point;

    if ( gis_mdp_set_center_point_ptr( mdp_ctx, &new_center ) != EOK ) {
        printf( "Failed to set center\n" );
        return 1;
    }

    if ( gis_mdp_get_center_point_deg( mdp_ctx, &point ) != EOK ) {
        printf( "Failed to get center point\n" );
        return 1;
    }

    if ( point.x != 40 && point.y != 45 ) {
        printf( "Incorrect center point\n" );
        return 1;
    }

```

Классификация:

ПК ЦКИ для ЗОСРВ «Нейтрино»

Безопасность использования	
Обработчик прерываний	Нет
Обработчик сигналов	Нет
В потоке	Да

Тематические ссылки:

[gis_mdp_t, gis_mdp_set_display_resolution\(\)](#)

gis_mdp_set_display_resolution()

Установка разрешения экрана для текущего отображения карты.

Прототип:

```
#include <gis/gis_mdp.h>

int gis_mdp_set_display_resolution( gis_mdp_t mdp_ctx, uint32_t width_px, uint32_t
↪ height_px );
```

Аргументы:

mdp_ctx

Контекст `gis_mdp_t`.

width_px

Ширина дисплея в пикселях.

height_px

Высота дисплея в пикселях.

Библиотека:

gishelper

Описание:

Функция устанавливает разрешение экрана для текущего отображения карты.

Возвращаемое значение:

EOK

Успешное завершение.

EINVAL

Некорректные параметры функции.

Примеры использования:

```

gis_mdp_t mdp_ctx;
gis_data_engine_context_t ctx;
gis_core_connection_t connection;
gis_core_link_init( &connection );
gis_core_link_connect( &connection, 777 );
gis_core_databuffer_attach( &connection );

uint32_t width = 600;
uint32_t height = 600;

gis_data_engine_alloc( width, height, GIS_DATA_ENGINE_MODE_SM, GIS_DATA_ENGINE_BPP_32,
↳ &connection, &ctx );

mdp_ctx = gis_data_engine_get_display_parameters( ctx );

if ( gis_mdp_set_display_resolution( mdp_ctx, 800, 600 ) != EOK ) {
    printf( "Failed to set resolution\n" );
    return 1;
}

int width_res = 600;
int height_res = 600;

gis_mdp_get_display_resolution( mdp_ctx, &width_res, &height_res );

if ( width_res != 800 && height_res != 600 )
{
    printf( "Incorrect resolution: %d %d\n", width_res, height_res );
    return 1;
}

```

Классификация:

ПК ЦКИ для ЗОСРВ «Нейтрино»

Безопасность использования	
Обработчик прерываний	Нет
Обработчик сигналов	Нет
В потоке	Да

Тематические ссылки:

[gis_mdp_t](#), [gis_mdp_set_display_size\(\)](#)

gis_mdp_set_display_size()

Установка геометрических размеров экрана для текущего отображения карты.

Прототип:

```
#include <gis/gis_mdp.h>

int gis_mdp_set_display_size( gis_mdp_t mdp_ctx, double width_mm, double height_mm );
```

Аргументы:

mdp_ctx

Контекст `gis_mdp_t`.

width_mm

Ширина дисплея в миллиметрах.

height_mm

Высота дисплея в миллиметрах.

Библиотека:

gishelper

Описание:

Функция устанавливает геометрические размеры экрана для текущего отображения карты.

Возвращаемое значение:

EOK

Успешное завершение.

EINVAL

Некорректные параметры функции.

Примеры использования:

```

gis_mdp_t mdp_ctx;
gis_data_engine_context_t ctx;
gis_core_connection_t connection;
gis_core_link_init( &connection );
gis_core_link_connect( &connection, 777 );
gis_core_databuffer_attach( &connection );

uint32_t width = 600;
uint32_t height = 600;

gis_data_engine_alloc( width, height, GIS_DATA_ENGINE_MODE_SM, GIS_DATA_ENGINE_BPP_32,
↳ &connection, &ctx );

mdp_ctx = gis_data_engine_get_display_parameters( ctx );

double width_res = 0;
double height_res = 0;

if ( gis_mdp_set_display_size( mdp_ctx, 510, 290 ) != EOK ) {
    printf( "Failed to set display size\n" );
    return 1;
}

gis_mdp_get_display_size( mdp_ctx, &width_res, &height_res );

if ( width_res != 510 && height_res != 290 )
{
    printf( "Incorrect display size: %f %f\n", width_res, height_res );
    return 1;
}

```

Классификация:

ПК ЦКИ для ЗОСРВ «Нейтрино»

Безопасность использования	
Обработчик прерываний	Нет
Обработчик сигналов	Нет
В потоке	Да

Тематические ссылки:

[gis_mdp_t](#), [gis_mdp_set_display_resolution\(\)](#), [gis_mdp_set_phys_scale\(\)](#)

gis_mdp_set_map_antialiasing_level()

Установка уровня сглаживания карты.

Прототип:

```
#include <gis/gis_mdp.h>

int gis_mdp_set_map_antialiasing_level( gis_mdp_t mdp_ctx, int level );
```

Аргументы:

mdp_ctx

Контекст `gis_mdp_t`.

level

Уровень сглаживания. Доступны следующие варианты: 0 — Без сглаживания. 1 — Сглаживание включено.

Библиотека:

gishelper

Описание:

Функция задаёт уровень сглаживания отображения карты.

Возвращаемое значение:

EOK

Успешное завершение.

EINVAL

Некорректные параметры функции.

Примеры использования:

```

gis_mdp_t mdp_ctx;
gis_data_engine_context_t ctx;
gis_core_connection_t connection;
gis_core_link_init( &connection );
gis_core_link_connect( &connection, 777 );
gis_core_databuffer_attach( &connection );

uint32_t width = 600;
uint32_t height = 600;

gis_data_engine_alloc( width, height, GIS_DATA_ENGINE_MODE_SM, GIS_DATA_ENGINE_BPP_32,
↳ &connection, &ctx );

mdp_ctx = gis_data_engine_get_display_parameters( ctx );

if ( gis_mdp_set_map_antialiasing_level( mdp_ctx, 1 ) != EOK ) {
    printf( "Failed to set antialiasing level\n" );
    return 1;
}

if ( gis_mdp_get_map_antialiasing_level( mdp_ctx ) != 1 ) {
    printf( "Incorrect antialiasing level\n" );
    return 1;
}

```

Классификация:

ПК ЦКИ для ЗОСРВ «Нейтрино»

Безопасность использования	
Обработчик прерываний	Нет
Обработчик сигналов	Нет
В потоке	Да

Поддерживается, начиная с ПК ЦКИ редакции 1.1.0

Тематические ссылки:

[gis_mdp_t](#), [gis_mdp_get_map_antialiasing_level\(\)](#)

gis_mdp_set_phys_scale()

Установка масштаба текущего отображения карты.

Прототип:

```
#include <gis/gis_mdp.h>

int gis_mdp_set_phys_scale( gis_mdp_t mdp_ctx, uint32_t real_scale_denom );
```

Аргументы:

mdp_ctx

Контекст `gis_mdp_t`.

real_scale_denom

Знаменатель масштаба (в метрах).

Библиотека:

gishelper

Описание:

Функция устанавливает масштаб текущего отображения карты как отношение 1:real_scale_denom.

Возвращаемое значение:

EOK

Успешное завершение.

EINVAL

Некорректные параметры функции.

Примеры использования:

```
gis_mdp_t mdp_ctx;
gis_data_engine_context_t ctx;
gis_core_connection_t connection;
gis_core_link_init( &connection );
gis_core_link_connect( &connection, 777 );
gis_core_databuffer_attach( &connection );

uint32_t width = 600;
uint32_t height = 600;
```

```

    gis_data_engine_alloc( width, height, GIS_DATA_ENGINE_MODE_SM, GIS_DATA_ENGINE_BPP_32,
↵  &connection, &ctx );

    mdp_ctx = gis_data_engine_get_display_parameters( ctx );

    if ( gis_mdp_set_phys_scale( mdp_ctx, 100000 ) != EOK ) {
        printf( "Failed to set scale\n" );
        return 1;
    }

    uint32_t scale = gis_mdp_get_phys_scale( mdp_ctx );

    if ( scale != 100000 ) {
        printf( "Incorrect scale: %d\n", scale );
        return 1;
    }

```

Классификация:

ПК ЦКИ для ЗОСРВ «Нейтрино»

Безопасность использования	
Обработчик прерываний	Нет
Обработчик сигналов	Нет
В потоке	Да

Тематические ссылки:

[gis_mdp_t](#), [gis_mdp_set_projection\(\)](#), [gis_mdp_get_phys_scale\(\)](#)

gis_mdp_set_projection()

Установка проекции текущего отображения карты.

Прототип:

```
#include <gis/gis_mdp.h>

int gis_mdp_set_projection( gis_mdp_t mdp_ctx, gis_map_projection_t *projection );
```

Аргументы:

mdp_ctx

Контекст `gis_mdp_t`.

parameters

Указатель на структуру параметров проекции тип `gis_map_projection_t`

Библиотека:

gishelper

Описание:

Функция устанавливает проекцию текущего отображения карты.

Возвращаемое значение:

EOK

Успешное завершение.

EINVAL

Некорректные параметры функции.

Примеры использования:

```
gis_mdp_t mdp_ctx;
gis_data_engine_context_t ctx;
gis_core_connection_t connection;
gis_core_link_init( &connection );
gis_core_link_connect( &connection, 777 );
gis_core_databuffer_attach( &connection );

uint32_t width = 600;
uint32_t height = 600;
```

```

    gis_data_engine_alloc( width, height, GIS_DATA_ENGINE_MODE_SM, GIS_DATA_ENGINE_BPP_32,
↪ &connection, &ctx );

    mdp_ctx = gis_data_engine_get_display_parameters( ctx );

    gis_map_projection_t projection;
    gis_map_projection_init( &projection );
    projection.projection_idx = GIS_PROJECTION_GAUSS_KRUGER_CONFORMAL;
    projection.ellipsoid_idx = GIS_ELLIPSOID_KRASSOVSKY_1942;
    projection.zone          = 7;

    if ( gis_mdp_set_projection( mdp_ctx, &projection ) != EOK ) {
        printf( "Failed to set projection\n" );
        return 1;
    }

    gis_map_projection_t *proj = gis_mdp_get_projection( mdp_ctx );
    if ( proj->projection_idx != GIS_PROJECTION_GAUSS_KRUGER_CONFORMAL &&
        proj->ellipsoid_idx  != GIS_ELLIPSOID_KRASSOVSKY_1942      &&
        proj->zone            != 7 )
    {
        printf( "Incorrect projection parameters\n" );
        return 1;
    }

```

Классификация:

ПК ЦКИ для ЗОСРВ «Нейтрино»

Безопасность использования	
Обработчик прерываний	Нет
Обработчик сигналов	Нет
В потоке	Да

Тематические ссылки:

[gis_mdp_t](#), [gis_mdp_get_projection\(\)](#), [gis_mdp_set_center_point\(\)](#)

gis_mdp_set_raster_color_mode()

Установка типа заливки для рисования растра.

Прототип:

```
#include <gis/gis_mdp.h>

int gis_mdp_set_raster_color_mode( gis_mdp_t mdp_ctx, bool discrete_flag );
```

Аргументы:

mdp_ctx

Контекст `gis_mdp_t`.

discrete_flag

Флаг, задающий требуемый тип рисования. Режимы работы: `false` — градиентная заливка. Используется по умолчанию. `true` — заливка дискретными цветами из палитры.

Библиотека:

gishelper

Описание:

Функция устанавливает тип заливки для рисования растровых файлов.

Возвращаемое значение:

EOK

Успешное завершение.

EINVAL

Некорректный указатель.

Примеры использования:

```

gis_mdp_t mdp_ctx;
gis_data_engine_context_t ctx;
gis_core_connection_t connection;
gis_core_link_init( &connection );
gis_core_link_connect( &connection, 777 );
gis_core_databuffer_attach( &connection );

uint32_t width = 600;
uint32_t height = 600;

gis_data_engine_alloc( width, height, GIS_DATA_ENGINE_MODE_SM, GIS_DATA_ENGINE_BPP_32,
↳ &connection, &ctx );

mdp_ctx = gis_data_engine_get_display_parameters( ctx );

gis_mdp_set_raster_color_mode( mdp_ctx, false );

if ( gis_mdp_get_raster_color_mode( mdp_ctx ) )
{
    printf( "Failed to set raster color mode\n" );
    return 1;
}

```

Классификация:

ПК ЦКИ для ЗОСРВ «Нейтрино»

Безопасность использования	
Обработчик прерываний	Нет
Обработчик сигналов	Нет
В потоке	Да

Тематические ссылки:

[gis_mdp_t](#), [gis_mdp_get_raster_color_mode\(\)](#) [gis_mdp_set_raster_height_mode\(\)](#) [gis_mdp_get_raster_height_mode\(\)](#)

gis_mdp_set_raster_height_limits()

Установка диапазона высот отображения для растровых файлов.

Прототип:

```
#include <gis/gis_mdp.h>

int gis_mdp_set_raster_height_limits( gis_mdp_t mdp_ctx, double lower, double upper );
```

Аргументы:

<i>mdp_ctx</i>	Контекст <code>gis_mdp_t</code> .
<i>lower</i>	Нижняя граница диапазона высот (в метрах).
<i>upper</i>	Верхняя граница диапазона высот (в метрах).

Библиотека:

gishelper

Описание:

Функция устанавливает предельные высоты отображения для растровых файлов.

Возвращаемое значение:

<i>EOK</i>	Успешное завершение.
<i>EINVAL</i>	Некорректный указатель.
<i>EFAULT</i>	Некорректный диапазон высот.

Примеры использования:

```

gis_mdp_t mdp_ctx;
gis_data_engine_context_t ctx;
gis_core_connection_t connection;
gis_core_link_init( &connection );
gis_core_link_connect( &connection, 777 );
gis_core_databuffer_attach( &connection );

uint32_t width = 600;
uint32_t height = 600;

gis_data_engine_alloc( width, height, GIS_DATA_ENGINE_MODE_SM, GIS_DATA_ENGINE_BPP_32,
↳ &connection, &ctx );

mdp_ctx = gis_data_engine_get_display_parameters( ctx );

gis_mdp_set_raster_height_limits( mdp_ctx, 40, 200 );

double upper_limit = gis_mdp_get_raster_upper_height_limit( mdp_ctx );
double lower_limit = gis_mdp_get_raster_lower_height_limit( mdp_ctx );

if ( upper_limit != 200 && lower_limit != 40 )
{
    printf( "Incorrect height limits\n" );
    return 1;
}

```

Классификация:

ПК ЦКИ для ЗОСРВ «Нейтрино»

Безопасность использования	
Обработчик прерываний	Нет
Обработчик сигналов	Нет
В потоке	Да

Тематические ссылки:

[gis_mdp_t](#), [gis_mdp_set_raster_color_mode\(\)](#), [gis_mdp_get_raster_lower_height_limit\(\)](#)

gis_mdp_set_raster_height_mode()

Установка режима выбора границ высот для растровых файлов.

Прототип:

```
#include <gis/gis_mdp.h>

void gis_mdp_set_raster_height_mode( gis_mdp_t mdp_ctx, bool auto_mode );
```

Аргументы:

mdp_ctx

Контекст `gis_mdp_t`.

auto_mode

Флаг, задающий режим выбора высот. Режимы работы: `false` — ручная установка диапазона высот. `true` — автоматическая настройка диапазона высот по доступным картам. Используется по умолчанию.

Библиотека:

gishelper

Описание:

Функция устанавливает режим выбора границ высот для отображения растровых файлов.

Примеры использования:

```
gis_mdp_t mdp_ctx;
gis_data_engine_context_t ctx;
gis_core_connection_t connection;
gis_core_link_init( &connection );
gis_core_link_connect( &connection, 777 );
gis_core_databuffer_attach( &connection );

uint32_t width = 600;
uint32_t height = 600;

gis_data_engine_alloc( width, height, GIS_DATA_ENGINE_MODE_SM, GIS_DATA_ENGINE_BPP_32,
↵ &connection, &ctx );

mdp_ctx = gis_data_engine_get_display_parameters( ctx );
```

```
gis_mdp_set_raster_height_mode( mdp_ctx, false );

if ( gis_mdp_get_raster_height_mode( mdp_ctx ) != false )
{
    printf( "Incorrect height mode\n" );
    return 1;
}
```

Классификация:

ПК ЦКИ для ЗОСРВ «Нейтрино»

Безопасность использования	
Обработчик прерываний	Нет
Обработчик сигналов	Нет
В потоке	Да

Тематические ссылки:

[gis_mdp_t](#), [gis_mdp_get_raster_height_mode\(\)](#)

gis_mdp_set_raster_invalid_height_color()

Установка цвета отображения высот вне заданного диапазона.

Прототип:

```
#include <gis/gis_mdp.h>

void gis_mdp_set_raster_invalid_height_color( gis_mdp_t mdp_ctx, uint32_t
↪ invalid_height_color );
```

Аргументы:

mdp_ctx

Контекст `gis_mdp_t`.

invalid_height_color

Цвет отображения высот (в формате ARGB), выходящих за пределы, настроенные при помощи функции `gis_mdp_set_raster_height_limits()`.

Библиотека:

gishelper

Описание:

Функция устанавливает цвет отображения высот вне заданного диапазона для растровых файлов.

Примеры использования:

```
gis_mdp_t mdp_ctx;
gis_data_engine_context_t ctx;
gis_core_connection_t connection;
gis_core_link_init( &connection );
gis_core_link_connect( &connection, 777 );
gis_core_databuffer_attach( &connection );

uint32_t width = 600;
uint32_t height = 600;

gis_data_engine_alloc( width, height, GIS_DATA_ENGINE_MODE_SM, GIS_DATA_ENGINE_BPP_32,
↪ &connection, &ctx );

mdp_ctx = gis_data_engine_get_display_parameters( ctx );
```

```
gis_mdp_set_raster_invalid_height_color( mdp_ctx, 0xFF00EE );

if ( gis_mdp_get_raster_invalid_height_color( mdp_ctx ) != 0xFF00EE )
{
    printf( "Failed to set invalid color value\n" );
    return 1;
}
```

Классификация:

ПК ЦКИ для ЗОСРВ «Нейтрино»

Безопасность использования	
Обработчик прерываний	Нет
Обработчик сигналов	Нет
В потоке	Да

Тематические ссылки:

`gis_mdp_t`, `gis_mdp_get_raster_invalid_height_color()` `gis_mdp_set_raster_color_mode()`, `gis_mdp_get_raster_lower_h`

Поддерживается, начиная с ПК ЦКИ редакции 1.1.0

gis_mdp_set_raster_palette()

Установка палитры для рисования раstra.

Прототип:

```
#include <gis/gis_mdp.h>

int gis_mdp_set_raster_palette( gis_mdp_t mdp_ctx, const uint32_t *colors, uint32_t
↪ color_number );
```

Аргументы:

mdp_ctx

Контекст `gis_mdp_t`.

colors

Указатель на массив цветовой палитры.

color_number

Количество цветов в массиве цветовой палитры.

Библиотека:

gishelper

Описание:

Функция устанавливает палитру рисования растровых карт.

Возвращаемое значение:

EOK

Успешное завершение.

EINVAL

Некорректный указатель.

ENOMEM

Не удалось выделить память для хранения палитры.

EBADR

Не удалась инициализация палитры.

Примеры использования:

```

gis_mdp_t mdp_ctx;
gis_data_engine_context_t ctx;
gis_core_connection_t connection;
gis_core_link_init( &connection );
gis_core_link_connect( &connection, 777 );
gis_core_databuffer_attach( &connection );

uint32_t width = 600;
uint32_t height = 600;

gis_data_engine_alloc( width, height, GIS_DATA_ENGINE_MODE_SM, GIS_DATA_ENGINE_BPP_32,
↳ &connection, &ctx );

mdp_ctx = gis_data_engine_get_display_parameters( ctx );

vector<uint32_t> colorMap;
colorMap.clear();
colorMap.push_back( 0x00004030 );
colorMap.push_back( 0x00186424 );
colorMap.push_back( 0x0048AC0C );
colorMap.push_back( 0x00ACC800 );
colorMap.push_back( 0x00D0E000 );
colorMap.push_back( 0x00F4F800 );
colorMap.push_back( 0x00E4C400 );
colorMap.push_back( 0x00D29A00 );
colorMap.push_back( 0x00C07000 );
colorMap.push_back( 0x00AE4600 );
colorMap.push_back( 0x00A22A00 );

if ( gis_mdp_set_raster_palette( mdp_ctx, colorMap.data(), colorMap.size() ) != EOK )
{
    printf( "Failed to set palette\n" );
    return 1;
}

uint32_t *get_palette;
uint32_t number_of_colors;
if ( ( get_palette = gis_mdp_get_raster_palette( mdp_ctx, &number_of_colors ) ) != NULL
↳ )
{
    if ( *get_palette != 0x00004030 ||
        *(get_palette + 1) != 0x00186424 )
    {
        printf( "Failed to get palette\n" );
        return 1;
    }
}

free( get_palette );

```

Классификация:

ПК ЦКИ для ЗОСРВ «Нейтрино»

Безопасность использования	
Обработчик прерываний	Нет
Обработчик сигналов	Нет
В потоке	Да

Тематические ссылки:

`gis_mdp_t`, `gis_mdp_get_raster_palette()`, `gis_mdp_set_raster_color_mode()`

gis_mdp_set_scaling_mode()

Установка режима масштабирования для текущего отображения карты.

Прототип:

```
#include <gis/gis_mdp.h>

int gis_mdp_set_scaling_mode( gis_mdp_t mdp_ctx, bool constant_scale );
```

Аргументы:

mdp_ctx

Контекст `gis_mdp_t`.

constant_scale

Режим масштабирования. true - картографический режим. В этом режиме толщина линий (в пикселях) постоянна. false - чертежный режим.

Библиотека:

gishelper

Описание:

Функция устанавливает режим масштабирования для текущего отображения карты.

Возвращаемое значение:

EOK

Успешное завершение.

EINVAL

Некорректный контекст `gis_mdp_t`.

Примеры использования:

```

gis_mdp_t mdp_ctx;
gis_data_engine_context_t ctx;
gis_core_connection_t connection;
gis_core_link_init( &connection );
gis_core_link_connect( &connection, 777 );
gis_core_databuffer_attach( &connection );

uint32_t width = 600;
uint32_t height = 600;

gis_data_engine_alloc( width, height, GIS_DATA_ENGINE_MODE_SM, GIS_DATA_ENGINE_BPP_32,
↳ &connection, &ctx );

mdp_ctx = gis_data_engine_get_display_parameters( ctx );

if ( gis_mdp_set_scaling_mode( mdp_ctx, true ) != EOK ) {
    printf( "Failed to set mode\n" );
    return 1;
}

if ( gis_mdp_get_scaling_mode( mdp_ctx ) != true ) {
    printf( "Invalid mode\n" );
    return 1;
}

```

Классификация:

ПК ЦКИ для ЗОСРВ «Нейтрино»

Безопасность использования	
Обработчик прерываний	Нет
Обработчик сигналов	Нет
В потоке	Да

Поддерживается, начиная с ПК ЦКИ редакции 1.0.0

Тематические ссылки:

[gis_mdp_t](#), [gis_mdp_get_scaling_mode\(\)](#)

gis_map_ellipsoid_idx_t

Индекс эллипсоида картографической информации.

Формат:

```
#include <gis/gis_types.h>

typedef enum {
    GIS_ELLIPSOID_EMPTY           = 0,
    GIS_ELLIPSOID_KRASSOVSKY_1942,
    GIS_ELLIPSOID_KLARK_1880,
    GIS_ELLIPSOID_WGS84,
    GIS_ELLIPSOID_GRS80
} gis_map_ellipsoid_idx_t;
```

Описание:

Перечисление gis_map_ellipsoid_idx_t включают следующие значения:

- GIS_ELLIPSOID_EMPTY — незаполненный эллипсоид
- GIS_ELLIPSOID_KRASSOVSKY_1942 — эллипсоид Красовского
- GIS_ELLIPSOID_KLARK_1880 — эллипсоид Кларка 1880 г.
- GIS_ELLIPSOID_WGS84 — эллипсоид WGS 1984 г. (default)
- GIS_ELLIPSOID_GRS80 — эллипсоид GRS80

Классификация:

ПК ЦКИ для ЗОСРВ «Нейтрино»

Тематические ссылки:

[gis_map_projection_t](#)

gis_map_height_system_idx_t

Индекс системы высот картографической информации.

Формат:

```
#include <gis/gis_types.h>

typedef enum {
    GIS_HEIGHT_SYSTEM_EMPTY           = 0,
    GIS_HEIGHT_SYSTEM_BALTIC,
    GIS_HEIGHT_SYSTEM_AUSTRALIAN,
    GIS_HEIGHT_SYSTEM_ADRIATIC,
    GIS_HEIGHT_SYSTEM_ZERO_NORMAL,
    GIS_HEIGHT_SYSTEM_OSTEND,
    GIS_HEIGHT_SYSTEM_LA_MANCHE,
    GIS_HEIGHT_SYSTEM_BELFAST,
    GIS_HEIGHT_SYSTEM_MALIK_HEAD,
    GIS_HEIGHT_SYSTEM_DUBLIN,
    GIS_HEIGHT_SYSTEM_PIRAEUS,
    GIS_HEIGHT_SYSTEM_DUTCH,
    GIS_HEIGHT_SYSTEM_REYKJAVIC,
    GIS_HEIGHT_SYSTEM_ALIKANTE,
    GIS_HEIGHT_SYSTEM_CANARY,
    GIS_HEIGHT_SYSTEM_GENOA,
    GIS_HEIGHT_SYSTEM_AMSTERDAM,
    GIS_HEIGHT_SYSTEM_OSLO,
    GIS_HEIGHT_SYSTEM_NARVIK,
    GIS_HEIGHT_SYSTEM_CASCAIS,
    GIS_HEIGHT_SYSTEM_HELSINKI,
    GIS_HEIGHT_SYSTEM_SWEDEN,
    GIS_HEIGHT_SYSTEM_MARSEILLE,
    GIS_HEIGHT_SYSTEM_TURKEY,
    GIS_HEIGHT_SYSTEM_USA_CANADA,
    GIS_HEIGHT_SYSTEM_BALTIC_1977,
    GIS_HEIGHT_SYSTEM_PACIFIC_OCEAN,
    GIS_HEIGHT_SYSTEM_WORLD_OCEAN,
} gis_map_height_system_idx_t;
```

Описание:

Перечисление gis_map_height_system_idx_t включают следующие значения:

- GIS_HEIGHT_SYSTEM_EMPTY — незаполненная система высот
- GIS_HEIGHT_SYSTEM_BALTIC — Балтийская система высот (от нуля Кронштадтского футштока)
- GIS_HEIGHT_SYSTEM_AUSTRALIAN — Австралийская система высот 1971 г.

- GIS_HEIGHT_SYSTEM_ADRIATIC — средний уровень Адриатического моря в Триесте
- GIS_HEIGHT_SYSTEM_ZERO_NORMAL — средний уровень Северного моря в Остенде «Зеро-Нормаль»
- GIS_HEIGHT_SYSTEM_OSTEND — средний уровень низких вод Северного моря в Остенде
- GIS_HEIGHT_SYSTEM_LA_MANCHE — средний уровень моря в проливе Ламанш
- GIS_HEIGHT_SYSTEM_BELFAST — средний уровень Ирландского моря в Белфасте
- GIS_HEIGHT_SYSTEM_MALIK_HEAD — средний уровень Атлантического океана в Малик-Хед
- GIS_HEIGHT_SYSTEM_DUBLIN — уровень низкой воды в Дублинском заливе
- GIS_HEIGHT_SYSTEM_PIRAEUS — средний уровень Эгейского моря в порту Пирей
- GIS_HEIGHT_SYSTEM_DUTCH — средний уровень моря у датского побережья
- GIS_HEIGHT_SYSTEM_REYKJAVIC — средний уровень залива Фахсафлоуи у Рейкьявика
- GIS_HEIGHT_SYSTEM_ALIKANTE — средний уровень Средиземного моря в Аликанте
- GIS_HEIGHT_SYSTEM_CANARY — средний уровень Атлантического океана у Канарских островов
- GIS_HEIGHT_SYSTEM_GENOA — средний уровень Лигурийского моря в Генуе
- GIS_HEIGHT_SYSTEM_AMSTERDAM — средний уровень Северного моря (нуль Амстердамского футштока)
- GIS_HEIGHT_SYSTEM_OSLO — средний уровень моря в Осло (Норвежский нормальный нуль)
- GIS_HEIGHT_SYSTEM_NARVIK — средний уровень моря в бухте Нарвик
- GIS_HEIGHT_SYSTEM_CASCAIS — средний уровень Атлантического океана в Кашкайш
- GIS_HEIGHT_SYSTEM_HELSINKI — средний уровень Балтийского моря в Хельсинки
- GIS_HEIGHT_SYSTEM_SWEDEN — средний уровень воды у шведских берегов
- GIS_HEIGHT_SYSTEM_MARSEILLE — средний уровень Средиземного моря в Марселе
- GIS_HEIGHT_SYSTEM_TURKEY — средний уровень морей, омывающих Турцию
- GIS_HEIGHT_SYSTEM_USA_CANADA — средний уровень морей и океанов, омывающих США и Канаду
- GIS_HEIGHT_SYSTEM_BALTIC_1977 — Балтийская система высот 1977 г.
- GIS_HEIGHT_SYSTEM_PACIFIC_OCEAN — средний уровень Охотского моря и Тихого океана
- GIS_HEIGHT_SYSTEM_WORLD_OCEAN — средний уровень мирового океана

Классификация:

ПК ЦКИ для ЗОСРВ «Нейтрино»

Тематические ссылки:

[gis_map_projection_t](#)

gis_map_projection_idx_t

Индекс проекции картографической информации.

Формат:

```
#include <gis/gis_types.h>

typedef enum {
    GIS_PROJECTION_EMPTY           = 0,
    GIS_PROJECTION_GAUSS_KRUGER_CONFORMAL,
    GIS_PROJECTION_CONIC_CONFORMAL,
    GIS_PROJECTION_UTM,
    GIS_PROJECTION_MILLER,
    GIS_PROJECTION_MERC_PSEUDO,
} gis_map_projection_idx_t;
```

Описание:

Перечисление `gis_map_projection_idx_t` включают следующие значения:

- `GIS_PROJECTION_EMPTY` — незаполненная проекция
- `GIS_PROJECTION_GAUSS_KRUGER_CONFORMAL` — проекция Гаусса-Крюгера
- `GIS_PROJECTION_CONIC_CONFORMAL` — коническая проекция
- `GIS_PROJECTION_UTM` — проекция UTM
- `GIS_PROJECTION_MILLER` — проекция Миллера
- `GIS_PROJECTION_MERC_PSEUDO` — проекция Меркатора (default)

Классификация:

ПК ЦКИ для ЗОСРВ «Нейтрино»

Тематические ссылки:

[gis_map_projection_t](#)

gis_map_projection_t

Расширенный контекст проекции карты.

Формат:

```
#include <gis/gis_types.h>

typedef struct {
    gis_map_projection_idx_t    projection_idx;
    gis_map_ellipsoid_idx_t    ellipsoid_idx;
    int32_t                    zone;
    gis_map_height_system_idx_t height_system_idx;
    int32_t                    isSouthHemisphere;
    int32_t                    EPSG;

    double                     first_main_parallel_degree;
    double                     second_main_parallel_degree;
    double                     axial_meridian_degree;
    double                     parallel_of_main_point;

    double                     false_northing;
    double                     false_easting;
} gis_map_projection_t;
```

Описание:

Структура gis_map_projection_t описывает расширенные параметры проекции, поля интерпретируются следующим образом:

Аргументы:

projection_idx

Индекс проекции ([gis_map_projection_idx_t](#)).

ellipsoid_idx

Индекс эллипсоида ([gis_map_ellipsoid_idx_t](#)).

zone

Номер зоны ([gis_map_projection_has_zone\(\)](#)).

height_system_idx

Индекс системы высот ([gis_map_height_system_idx_t](#)).

isSouthHemisphere

Флаг нахождения в южном полушарии.

EPSG

EPSG код проекции.

first_main_parallel_degree

Первая главная параллель (в градусах).

second_main_parallel_degree

Вторая главная параллель (в градусах).

axial_meridian_degree

Осевой меридиан (в градусах).

parallel_of_main_point

Параллель главной точки (в градусах).

false_northing

Смещение на север (в метрах).

false_easting

Смещение на восток (в метрах).

Классификация:

ПК ЦКИ для ЗОСРВ «Нейтрино»

Тематические ссылки:

`gis_map_projection_idx_t, gis_map_ellipsoid_idx_t, gis_map_height_system_idx_t, gis_mdp_set_projection(),
gis_map_projection_init()`

gis_mdp_t

Контекст параметров визуализации.

Формат:

```
#include <gis/gis_mdp.h>

typedef void *      gis_mdp_t;
```

Описание:

Данный контекст обеспечивает управление параметрами визуализации ЦКИ библиотеками gishelper и gisrender.

Классификация:

ПК ЦКИ для ЗОСРВ «Нейтрино»

gis_render_sm_alloc()

Создание контекста движка рендеринга.

Прототип:

```
#include <gis/gis_surfacemanager.h>

int gis_render_sm_alloc( gis_render_sm_context_t *sm_ctx, gis_data_engine_context_t ctx,
↳ uint32_t surface_extension_px, double blit_region_rate, uint32_t flags );
```

Аргументы:

sm_ctx

Указатель на контекст движка рендеринга Surface Manager.

ctx

Указатель на контекст движка рендеринга Data Engine.

surface_extension_px

Величина расширения поверхности относительно видимой области виджета в пикселях.

blit_region_rate

Коэффициент, задающий размер области блиттинга (зоны нечувствительности), при перемещении в которой не будет повторно рисоваться поверхность. Задаётся в пределах от 0 до 1.0. Режимы работы: 0 — Область блиттинга совпадает с областью поверхности. 1 — Область блиттинга совпадает с областью виджета.

flags

Флаги, определяющие режим работы Surface Manager. Возможные варианты: GIS_RENDER_S↳ M_ALLOC_RAM — поверхности для рендеринга будут выделены в RAM памяти.

Библиотека:

surfacemanager

Описание:

Функция создаёт контекст движка рендеринга Surface Manager и инициализирует его переданными параметрами.

Возвращаемое значение:

EOK

Успешное завершение

EINVAL

Некорректные параметры функции

EFAULT

Не удалось инициализировать контекст Surface Manager

ENOMEM

Недостаточно памяти

ENOENT

Некорректный режим движка рендеринга Data Engine

Примеры использования:

```
QApplication app(argc, argv);
gis_core_connection_t connection;
gis_core_request_parameters_t map;
gis_data_engine_context_t data_engine_ctx;
gis_render_sm_context_t sm_ctx;

gis_core_link_init( &connection );
gis_core_link_connect( &connection, 777 );
gis_core_databuffer_attach( &connection );
gis_core_request_parameters_init( &map );
gis_core_databuffer_data_request( &connection, &map, 1 );
gis_data_engine_alloc( 800, 600, GIS_DATA_ENGINE_MODE_SM, GIS_DATA_ENGINE_BPP_32,
↪ &connection, &data_engine_ctx );

if ( gis_render_sm_alloc( &sm_ctx, data_engine_ctx, 1000, 0.5, 0 ) != EOK )
{
    printf( "Failed to allocate sm context\n" );
    return 1;
}

gis_render_sm_free( &sm_ctx );
gis_data_engine_free( &data_engine_ctx );
gis_core_databuffer_detach();
gis_core_link_destroy( &connection, true );
```

Классификация:

ПК ЦКИ для ЗОСРВ «Нейтрино»

Безопасность использования	
Обработчик прерываний	Нет
Обработчик сигналов	Нет
В потоке	Да

Тематические ссылки:

[gis_render_sm_context_t](#), [gis_render_sm_move\(\)](#), [gis_render_sm_free\(\)](#)

gis_render_sm_calculate_distance()

Вычисление длины траектории, описываемой градусными точками.

Прототип:

```
#include <gis/gis_surfacemanager.h>

int gis_render_sm_calculate_distance( gis_render_sm_context_t sm_ctx, double_point_t
↪ *points, uint32_t n_points, double* distance );
```

Аргументы:

<i>sm_ctx</i>	Контекст движка рендеринга Surface Manager.
<i>points</i>	Массив точек.
<i>n_points</i>	Количество точек.
<i>distance</i>	Указатель на переменную, хранящую расстояние.

Библиотека:

surfacemanager

Описание:

Функция вычисляет расстояние длины траектории, описываемой градусными точками, на текущем эллипсоиде проекции отображения.

Возвращаемое значение:

<i>EOK</i>	Успешное завершение
<i>EINVAL</i>	Некорректные параметры функции

Примеры использования:

```

QApplication app(argc, argv);
gis_core_connection_t connection;
gis_core_request_parameters_t map;
gis_data_engine_context_t data_engine_ctx;
gis_render_sm_context_t sm_ctx;

gis_core_link_init( &connection );
gis_core_link_connect( &connection, 777 );
gis_core_databuffer_attach( &connection );
gis_core_request_parameters_init( &map );
gis_core_databuffer_data_request( &connection, &map, 1 );
gis_data_engine_alloc( 800, 600, GIS_DATA_ENGINE_MODE_SM, GIS_DATA_ENGINE_BPP_32,
↪ &connection, &data_engine_ctx );
gis_render_sm_alloc( &sm_ctx, data_engine_ctx, 1000, 0.5, 0 );

vector<double_point_t> degree_vector;
double distance = 0;
double_point_t pnt;
pnt.x = 34;
pnt.y = 36;
degree_vector.push_back( pnt );
pnt.x = 35;
pnt.y = 39;
degree_vector.push_back( pnt );

gis_render_sm_calculate_distance( sm_ctx, degree_vector.data(), degree_vector.size(),
↪ &distance );

if ( fabs( distance - 344496 ) < 0.1 )
{
    printf( "Incorrect distance: %lf", distance );
    return 1;
}

gis_render_sm_free( &sm_ctx );
gis_data_engine_free( &data_engine_ctx );
gis_core_databuffer_detach();
gis_core_link_destroy( &connection, true );

```

Классификация:

ПК ЦКИ для ЗОСРВ «Нейтрино»

Безопасность использования	
Обработчик прерываний	Нет
Обработчик сигналов	Нет
В потоке	Да

Тематические ссылки:

`gis_render_sm_context_t`, `gis_render_sm_calculate_polygon()`

gis_render_sm_calculate_polygon()

Вычисление параметров полигона.

Прототип:

```
#include <gis/gis_surfacemanager.h>

int gis_render_sm_calculate_polygon( gis_render_sm_context_t sm_ctx, double_point_t
↪ *points, uint32_t n_points, double *area, double *perimeter );
```

Аргументы:

sm_ctx

Контекст движка рендеринга Surface Manager.

points

Массив точек.

n_points

Количество точек.

area

Указатель на переменную, хранящую площадь полигона.

perimeter

Указатель на переменную, хранящую периметр полигона.

Библиотека:

surfacemanager

Описание:

Функция вычисляет площадь и периметр полигона, описанного градусными точками.

Возвращаемое значение:

EOK

Успешное завершение

EINVAL

Некорректные параметры функции

Примеры использования:

```

QApplication app(argc, argv);
gis_core_connection_t connection;
gis_core_request_parameters_t map;
gis_data_engine_context_t data_engine_ctx;
gis_render_sm_context_t sm_ctx;

gis_core_link_init( &connection );
gis_core_link_connect( &connection, 777 );
gis_core_databuffer_attach( &connection );
gis_core_request_parameters_init( &map );
gis_core_databuffer_data_request( &connection, &map, 1 );
gis_data_engine_alloc( 800, 600, GIS_DATA_ENGINE_MODE_SM, GIS_DATA_ENGINE_BPP_32,
↪ &connection, &data_engine_ctx );
gis_render_sm_alloc( &sm_ctx, data_engine_ctx, 1000, 0.5, 0 );

vector<double_point_t> degree_vector;
double area = 0, perimeter = 0;

double_point_t pnt;
pnt.x = 10;
pnt.y = 10;
degree_vector.push_back( pnt );
pnt.x = 10;
pnt.y = 11;
degree_vector.push_back( pnt );
pnt.x = 11;
pnt.y = 11;
degree_vector.push_back( pnt );
pnt.x = 11;
pnt.y = 10;
degree_vector.push_back( pnt );

gis_render_sm_calculate_polygon( sm_ctx, degree_vector.data(), degree_vector.size(),
↪ &area, &perimeter );

if ( fabs( perimeter - 440149 ) > 10 )
{
    printf( "gis_render_sm_calculate_polygon() failed\n" );
    return 1;
}

gis_render_sm_free( &sm_ctx );
gis_data_engine_free( &data_engine_ctx );
gis_core_databuffer_detach();
gis_core_link_destroy( &connection, true );

```

Классификация:

ПК ЦКИ для ЗОСРВ «Нейтрино»

Безопасность использования	
<i>Обработчик прерываний</i>	Нет
<i>Обработчик сигналов</i>	Нет
<i>В потоке</i>	Да

Тематические ссылки:

`gis_render_sm_context_t`, `gis_render_sm_calculate_distance()`

gis_render_sm_convert_degree2fpx()

Преобразование градусных координат точки в пиксельные координаты окна отображения с сохранением дробной части.

Прототип:

```
#include <gis/gis_surfacemanager.h>

int gis_render_sm_convert_degree2fpx( gis_render_sm_context_t sm_ctx, double_point_t
↪ degree_pos, double_point_t *fpx_pos );
```

Аргументы:

<i>sm_ctx</i>	Контекст движка рендеринга Surface Manager.
<i>degree_pos</i>	Точка с градусными координатами.
<i>fpx_pos</i>	Указатель на точку с пиксельными координатами.

Библиотека:

surfacemanager

Описание:

Функция преобразует градусные координаты точки в пиксельные координаты окна отображения с сохранением дробной части

Возвращаемое значение:

<i>EOK</i>	Успешное завершение
<i>EINVAL</i>	Некорректные параметры функции
<i>ENOENT</i>	Контекст Surface Manager не был инициализирован
<i>EFAULT</i>	Ошибка преобразования
<i>EDOM</i>	Некорректные градусные координаты

Примеры использования:

```
QApplication app(argc, argv);
gis_core_connection_t connection;
gis_core_request_parameters_t map;
gis_data_engine_context_t data_engine_ctx;
gis_render_sm_context_t sm_ctx;

gis_core_link_init( &connection );
gis_core_link_connect( &connection, 777 );
gis_core_databuffer_attach( &connection );
gis_core_request_parameters_init( &map );
gis_core_databuffer_data_request( &connection, &map, 1 );
gis_data_engine_alloc( 800, 600, GIS_DATA_ENGINE_MODE_SM, GIS_DATA_ENGINE_BPP_32,
↪ &connection, &data_engine_ctx );
gis_render_sm_alloc( &sm_ctx, data_engine_ctx, 1000, 0.5, 0 );

double_point_t px_pnt;
double_point_t degree_pos;
degree_pos.x = 45;
degree_pos.y = 40;

gis_render_sm_convert_degree2fpx( sm_ctx, degree_pos, &px_pnt );

if ( fabs( px_pnt.x - 11619.602222 ) < 0.1 &&
    fabs( px_pnt.y - 11277.539252 ) < 0.1 )
{
    printf( "Correct\n" );
}
else
{
    printf( "Incorrect px coords: %lf %lf\n", px_pnt.x, px_pnt.y );
    return 1;
}

gis_render_sm_free( &sm_ctx );
gis_data_engine_free( &data_engine_ctx );
gis_core_databuffer_detach();
gis_core_link_destroy( &connection, true );
```

Классификация:

ПК ЦКИ для ЗОСРВ «Нейтрино»

Безопасность использования	
Обработчик прерываний	Нет
Обработчик сигналов	Нет
В потоке	Да

Тематические ссылки:

`gis_render_sm_context_t`, `gis_render_sm_convert_degree2px()`

gis_render_sm_convert_degree2px()

Преобразование градусных координат точки в пиксельные координаты окна отображения.

Прототип:

```
#include <gis/gis_surfacemanager.h>

int gis_render_sm_convert_degree2px( gis_render_sm_context_t sm_ctx, double_point_t
↪ degree_pos, int32_point_t *px_pos );
```

Аргументы:

<i>sm_ctx</i>	Контекст движка рендеринга Surface Manager.
<i>degree_pos</i>	Точка с градусными координатами.
<i>px_pos</i>	Указатель на точку с пиксельными координатами.

Библиотека:

surfacemanager

Описание:

Функция преобразует градусные координаты точки в пиксельные координаты окна отображения

Возвращаемое значение:

<i>EOK</i>	Успешное завершение
<i>EINVAL</i>	Некорректные параметры функции
<i>ENOENT</i>	Контекст Surface Manager не был инициализирован
<i>EFAULT</i>	Ошибка преобразования
<i>EDOM</i>	Некорректные градусные координаты

Примеры использования:

```
QApplication app(argc, argv);
gis_core_connection_t connection;
gis_core_request_parameters_t map;
gis_data_engine_context_t data_engine_ctx;
gis_render_sm_context_t sm_ctx;

gis_core_link_init( &connection );
gis_core_link_connect( &connection, 777 );
gis_core_databuffer_attach( &connection );
gis_core_request_parameters_init( &map );
gis_core_databuffer_data_request( &connection, &map, 1 );
gis_data_engine_alloc( 800, 600, GIS_DATA_ENGINE_MODE_SM, GIS_DATA_ENGINE_BPP_32,
↪ &connection, &data_engine_ctx );
gis_render_sm_alloc( &sm_ctx, data_engine_ctx, 1000, 0.5, 0 );

int32_point_t px_pnt;
double_point_t degree_pos;
degree_pos.x = 45;
degree_pos.y = 40;

gis_render_sm_convert_degree2px( sm_ctx, degree_pos, &px_pnt );

if ( px_pnt.x != 11620 && px_pnt.y != 11278 )
{
    printf( "Incorrect px coords: %d %d\n", px_pnt.x, px_pnt.y );
    return 1;
}

gis_render_sm_free( &sm_ctx );
gis_data_engine_free( &data_engine_ctx );
gis_core_databuffer_detach();
gis_core_link_destroy( &connection, true );
```

Классификация:

ПК ЦКИ для ЗОСРВ «Нейтрино»

Безопасность использования	
Обработчик прерываний	Нет
Обработчик сигналов	Нет
В потоке	Да

Тематические ссылки:

[gis_render_sm_context_t](#), [gis_render_sm_convert_degree2fpx\(\)](#)

gis_render_sm_convert_fpx2degree()

Преобразование пиксельных координат точки окна отображения в градусные с сохранением дробной части.

Прототип:

```
#include <gis/gis_surfacemanager.h>

int gis_render_sm_convert_fpx2degree( gis_render_sm_context_t sm_ctx, double_point_t
↪ fpx_pos, double_point_t* degree_pos );
```

Аргументы:

<i>sm_ctx</i>	Контекст движка рендеринга Surface Manager.
<i>fpx_pos</i>	Точка с пиксельными координатами.
<i>degree_pos</i>	Указатель на точка с градусными координатами.

Библиотека:

surfacemanager

Описание:

Функция преобразует пиксельные координаты точки с сохранением дробной части в градусные

Возвращаемое значение:

<i>EOK</i>	Успешное завершение
<i>EINVAL</i>	Некорректные параметры функции
<i>ENOENT</i>	Контекст Surface Manager не был инициализирован
<i>EFAULT</i>	Ошибка преобразования

Примеры использования:

```

QApplication app(argc, argv);
gis_core_connection_t connection;
gis_core_request_parameters_t map;
gis_data_engine_context_t data_engine_ctx;
gis_render_sm_context_t sm_ctx;

gis_core_link_init( &connection );
gis_core_link_connect( &connection, 777 );
gis_core_databuffer_attach( &connection );
gis_core_request_parameters_init( &map );
gis_core_databuffer_data_request( &connection, &map, 1 );
gis_data_engine_alloc( 800, 600, GIS_DATA_ENGINE_MODE_SM, GIS_DATA_ENGINE_BPP_32,
↪ &connection, &data_engine_ctx );
gis_render_sm_alloc( &sm_ctx, data_engine_ctx, 1000, 0.5, 0 );

double_point_t degree_pnt;
double_point_t px_pos;
px_pos.x = 100.1;
px_pos.y = 103.2;
gis_render_sm_convert_fpx2degree( sm_ctx, px_pos, &degree_pnt );

if ( fabs( degree_pnt.x - 39.866350 ) < 0.0001 &&
    fabs( degree_pnt.y - 43.730465 ) < 0.0001 )
{
    printf( "Correct\n" );
}
else
{
    printf( "Incorrect degrees: %lf %lf\n", degree_pnt.x, degree_pnt.y );
    return 1;
}

gis_render_sm_free( &sm_ctx );
gis_data_engine_free( &data_engine_ctx );
gis_core_databuffer_detach();
gis_core_link_destroy( &connection, true );

```

Классификация:

ПК ЦКИ для ЗОСРВ «Нейтрино»

Безопасность использования	
Обработчик прерываний	Нет
Обработчик сигналов	Нет
В потоке	Да

Тематические ссылки:

```
gis_render_sm_context_t, gis_render_sm_convert_degree2px()
```

gis_render_sm_convert_fpx2meters()

Преобразование пиксельных координат точки окна отображения в метрические координаты с сохранением дробной части.

Прототип:

```
#include <gis/gis_surfacemanager.h>

int gis_render_sm_convert_fpx2meters( gis_render_sm_context_t sm_ctx, double_point_t
↪ fpx_pos, double_point_t *meters_pos );
```

Аргументы:

<i>sm_ctx</i>	Контекст движка рендеринга Surface Manager.
<i>fpx_pos</i>	Точка с пиксельными координатами.
<i>meters_pos</i>	Указатель на точку с метрическими координатами.

Библиотека:

surfacemanager

Описание:

Функция преобразует пиксельные координаты точки с сохранением дробной части в метрические координаты

Возвращаемое значение:

<i>EOK</i>	Успешное завершение
<i>EINVAL</i>	Некорректные параметры функции
<i>ENOENT</i>	Контекст Surface Manager не был инициализирован
<i>EFAULT</i>	Ошибка преобразования

Примеры использования:

```

QApplication app(argc, argv);
gis_core_connection_t connection;
gis_core_request_parameters_t map;
gis_data_engine_context_t data_engine_ctx;
gis_render_sm_context_t sm_ctx;

gis_core_link_init( &connection );
gis_core_link_connect( &connection, 777 );
gis_core_databuffer_attach( &connection );
gis_core_request_parameters_init( &map );
gis_core_databuffer_data_request( &connection, &map, 1 );
gis_data_engine_alloc( 800, 600, GIS_DATA_ENGINE_MODE_SM, GIS_DATA_ENGINE_BPP_32,
↪ &connection, &data_engine_ctx );
gis_render_sm_alloc( &sm_ctx, data_engine_ctx, 1000, 0.5, 0 );

double_point_t px_pnt;
px_pnt.x = 100;
px_pnt.y = 140;
double_point_t meter_point;

gis_render_sm_convert_fpx2meters( sm_ctx, px_pnt, &meter_point );

if ( fabs( meter_point.x - 4437896.819231 ) < 0.1 &&
    fabs( meter_point.y - 5392447.959814 ) < 0.1 )
{
    printf( "Correct\n" );
}
else
{
    printf( "Incorrect meters coords: %lf %lf\n", meter_point.x, meter_point.y );
    gis_render_sm_free( &sm_ctx );
    gis_data_engine_free( &data_engine_ctx );
    gis_core_databuffer_detach();
    gis_core_link_destroy( &connection, true );
    return 1;
}

gis_render_sm_free( &sm_ctx );
gis_data_engine_free( &data_engine_ctx );
gis_core_databuffer_detach();
gis_core_link_destroy( &connection, true );

```

Классификация:

ПК ЦКИ для ЗОСРВ «Нейтрино»

Безопасность использования	
<i>Обработчик прерываний</i>	Нет
<i>Обработчик сигналов</i>	Нет
<i>В потоке</i>	Да

Тематические ссылки:

`gis_render_sm_context_t`, `gis_render_sm_convert_degree2px()`

gis_render_sm_convert_px2degree()

Преобразование пиксельных координат точки окна отображения в градусные.

Прототип:

```
#include <gis/gis_surfacemanager.h>

int gis_render_sm_convert_px2degree( gis_render_sm_context_t sm_ctx, int32_point_t px_pos,
↪ double_point_t* degree_pos );
```

Аргументы:

<i>sm_ctx</i>	Контекст движка рендеринга Surface Manager.
<i>px_pos</i>	Точка с пиксельными координатами.
<i>degree_pos</i>	Указатель на точка с градусными координатами.

Библиотека:

surfacemanager

Описание:

Функция преобразует пиксельные координаты точки окна отображения в градусные

Возвращаемое значение:

<i>EOK</i>	Успешное завершение
<i>EINVAL</i>	Некорректные параметры функции
<i>ENOENT</i>	Контекст Surface Manager не был инициализирован
<i>EFAULT</i>	Ошибка преобразования

Примеры использования:

```

QApplication app(argc, argv);
gis_core_connection_t connection;
gis_core_request_parameters_t map;
gis_data_engine_context_t data_engine_ctx;
gis_render_sm_context_t sm_ctx;

gis_core_link_init( &connection );
gis_core_link_connect( &connection, 777 );
gis_core_databuffer_attach( &connection );
gis_core_request_parameters_init( &map );
gis_core_databuffer_data_request( &connection, &map, 1 );
gis_data_engine_alloc( 800, 600, GIS_DATA_ENGINE_MODE_SM, GIS_DATA_ENGINE_BPP_32,
↪ &connection, &data_engine_ctx );
gis_render_sm_alloc( &sm_ctx, data_engine_ctx, 1000, 0.5, 0 );

double_point_t degree_pnt;
int32_point_t px_pos;
px_pos.x = 100;
px_pos.y = 150;
gis_render_sm_convert_px2degree( sm_ctx, px_pos, &degree_pnt );

if ( fabs( degree_pnt.x - 39.866305 ) < 0.0001 &&
    fabs( degree_pnt.y - 43.715300 ) < 0.0001 )
{
    printf( "Correct\n" );
}
else
{
    printf( "Incorrect degrees: %lf %lf\n", degree_pnt.x, degree_pnt.y );
    return 1;
}

gis_render_sm_free( &sm_ctx );
gis_data_engine_free( &data_engine_ctx );
gis_core_databuffer_detach();
gis_core_link_destroy( &connection, true );

```

Классификация:

ПК ЦКИ для ЗОСРВ «Нейтрино»

Безопасность использования	
Обработчик прерываний	Нет
Обработчик сигналов	Нет
В потоке	Да

Тематические ссылки:

```
gis_render_sm_context_t, gis_render_sm_convert_degree2px()
```

gis_render_sm_convert_px2meters()

Преобразование пиксельных координат точки окна отображения в метрические координаты.

Прототип:

```
#include <gis/gis_surfacemanager.h>

int gis_render_sm_convert_px2meters( gis_render_sm_context_t sm_ctx, int32_point_t px_pos,
↪ double_point_t *meters_pos );
```

Аргументы:

<i>sm_ctx</i>	Контекст движка рендеринга Surface Manager.
<i>px_pos</i>	Точка с пиксельными координатами.
<i>meters_pos</i>	Указатель на точку с метрическими координатами.

Библиотека:

surfacemanager

Описание:

Функция преобразует пиксельные координаты точки окна отображения в метрические координаты

Возвращаемое значение:

<i>EOK</i>	Успешное завершение
<i>EINVAL</i>	Некорректные параметры функции
<i>ENOENT</i>	Контекст Surface Manager не был инициализирован
<i>EFAULT</i>	Ошибка преобразования

Примеры использования:

```

QApplication app(argc, argv);
gis_core_connection_t connection;
gis_core_request_parameters_t map;
gis_data_engine_context_t data_engine_ctx;
gis_render_sm_context_t sm_ctx;

gis_core_link_init( &connection );
gis_core_link_connect( &connection, 777 );
gis_core_databuffer_attach( &connection );
gis_core_request_parameters_init( &map );
gis_core_databuffer_data_request( &connection, &map, 1 );
gis_data_engine_alloc( 800, 600, GIS_DATA_ENGINE_MODE_SM, GIS_DATA_ENGINE_BPP_32,
↪ &connection, &data_engine_ctx );
gis_render_sm_alloc( &sm_ctx, data_engine_ctx, 1000, 0.5, 0 );

int32_point_t px_pnt;
px_pnt.x = 100;
px_pnt.y = 140;
double_point_t meter_point;

gis_render_sm_convert_px2meters( sm_ctx, px_pnt, &meter_point );

if ( fabs( meter_point.x - 4437896.819231 ) < 0.1 &&
    fabs( meter_point.y - 5392447.959814 ) < 0.1 )
{
    printf( "Correct\n" );
}
else
{
    printf( "Incorrect meters coords: %lf %lf\n", meter_point.x, meter_point.y );
    return 1;
}

gis_render_sm_free( &sm_ctx );
gis_data_engine_free( &data_engine_ctx );
gis_core_databuffer_detach();
gis_core_link_destroy( &connection, true );

```

Классификация:

ПК ЦКИ для ЗОСРВ «Нейтрино»

Безопасность использования	
Обработчик прерываний	Нет
Обработчик сигналов	Нет
В потоке	Да

Тематические ссылки:

`gis_render_sm_context_t`, `gis_render_sm_convert_degree2px()`

gis_render_sm_draw()

Вызов обработчика рисования движка рендеринга *Surface Manager*.

Прототип:

```
#include <gis/gis_surfacemanager.h>

int gis_render_sm_draw( gis_render_sm_context_t sm_ctx );
```

Аргументы:

sm_ctx

Контекст движка рендеринга *Surface Manager*.

Библиотека:

surfacemanager

Описание:

Функция вызывает обработчик рисования движка рендеринга *Surface Manager*.

Возвращаемое значение:

EOK

Успешное завершение

EINVAL

Некорректные параметры функции

ENOENT

Контекст *Surface Manager* не был инициализирован

Примеры использования:

```
gis_render_sm_context_t    sm_ctx;

void MapWidget::paintEvent(QPaintEvent * event)
{
    gis_render_sm_draw( sm_ctx );
    QPixmap pixmap = *(QPixmap *)gis_render_sm_get_view_pixmap( sm_ctx, &viewport_pos );
    QPainter painter( this );
    painter.drawPixmap( 0, 0, pixmap, viewport_pos.x, viewport_pos.y, width(), height()
↵ );
}
```

Классификация:

ПК ЦКИ для ЗОСРВ «Нейтрино»

Безопасность использования	
Обработчик прерываний	Нет
Обработчик сигналов	Нет
В потоке	Да

Поддерживается, начиная с ПК ЦКИ редакции 1.0.0

Поддерживается, начиная с ПК ЦКИ редакции 1.2

Тематические ссылки:

`gis_render_sm_context_t`, `gis_render_sm_move()`, `gis_render_sm_get_view_pixmap()`

gis_render_sm_free()

Освобождение контекста движка рендеринга Surface Manager.

Прототип:

```
#include <gis/gis_surfacemanager.h>

int gis_render_sm_free( gis_render_sm_context_t *sm_ctx );
```

Аргументы:

sm_ctx

Указатель на контекст движка рендеринга Surface Manager.

Библиотека:

surfacemanager

Описание:

Функция освобождает контекст движка рендеринга Surface Manager.

Возвращаемое значение:

EOK

Успешное завершение

EINVAL

Некорректные параметры функции

EALREADY

Контекст Surface Manager не был инициализирован

Примеры использования:

```
QApplication app(argc, argv);
gis_core_connection_t connection;
gis_core_request_parameters_t map;
gis_data_engine_context_t data_engine_ctx;
gis_render_sm_context_t sm_ctx;

gis_core_link_init( &connection );
gis_core_link_connect( &connection, 777 );
gis_core_databuffer_attach( &connection );
```

```

gis_core_request_parameters_init( &map );
gis_core_databuffer_data_request( &connection, &map, 1 );
gis_data_engine_alloc( 800, 600, GIS_DATA_ENGINE_MODE_SM, GIS_DATA_ENGINE_BPP_32,
↪ &connection, &data_engine_ctx );
gis_render_sm_alloc( &sm_ctx, data_engine_ctx, 1000, 0.5, 0 );

if ( gis_render_sm_free( &sm_ctx ) != EOK ) {
    printf( "gis_render_sm_free() failed.\n" );
    return 1;
}

gis_data_engine_free( &data_engine_ctx );
gis_core_databuffer_detach();
gis_core_link_destroy( &connection, true );

```

Классификация:

ПК ЦКИ для ЗОСРВ «Нейтрино»

Безопасность использования	
Обработчик прерываний	Нет
Обработчик сигналов	Нет
В потоке	Да

Тематические ссылки:

[gis_render_sm_context_t](#), [gis_render_sm_move\(\)](#), [gis_render_sm_alloc\(\)](#)

gis_render_sm_get_draw_canvas_borders()

Получение значений границ холста в градусах.

Прототип:

```
#include <gis/gis_surfacemanager.h>

int gis_render_sm_get_draw_canvas_borders( gis_render_sm_context_t sm_ctx, gis_borders_t
↵ *canvas_borders );
```

Аргументы:

sm_ctx

Контекст движка рендеринга Surface Manager.

canvas_borders

Указатель на границы `gis_borders_t`.

Библиотека:

surfacemanager

Описание:

Функция заполняет поля передаваемой структуры текущими значениями границ холста в градусах.

Возвращаемое значение:

EOK

Успешное завершение

EINVAL

Некорректные параметры функции

Примеры использования:

```

QApplication app(argc, argv);
gis_core_connection_t      connection;
gis_core_request_parameters_t map;
gis_data_engine_context_t  data_engine_ctx;
gis_render_sm_context_t    sm_ctx;

gis_core_link_init( &connection );
gis_core_link_connect( &connection, 777 );
gis_core_databuffer_attach( &connection );
gis_core_request_parameters_init( &map );
gis_core_databuffer_data_request( &connection, &map, 1 );
gis_data_engine_alloc( 800, 600, GIS_DATA_ENGINE_MODE_SM, GIS_DATA_ENGINE_BPP_32,
↪ &connection, &data_engine_ctx );
gis_render_sm_alloc( &sm_ctx, data_engine_ctx, 1000, 0.5, 0 );

gis_borders_t canvas_borders;

int res = gis_render_sm_get_draw_canvas_borders( sm_ctx, &canvas_borders );
if ( res != EOK )
{
    printf( "Failed to get canvas borders.\n" );
    return 1;
}

gis_render_sm_free( &sm_ctx );
gis_data_engine_free( &data_engine_ctx );
gis_core_databuffer_detach();
gis_core_link_destroy( &connection, true );

```

Классификация:

ПК ЦКИ для ЗОСРВ «Нейтрино»

Безопасность использования	
Обработчик прерываний	Нет
Обработчик сигналов	Нет
В потоке	Да

Поддерживается, начиная с ПК ЦКИ редакции 1.4

Тематические ссылки:

[gis_render_sm_context_t](#), [gis_borders_t](#)

gis_render_sm_get_layer_type()

Получение типа слоя, назначенного указанной поверхности.

Прототип:

```
#include <gis/gis_surfacemanager.h>

gis_render_sm_layer_type_t gis_render_sm_get_layer_type( gis_render_sm_context_t sm_ctx,
↳ uint8_t idx );
```

Аргументы:

sm_ctx

Контекст движка рендеринга Surface Manager.

idx

Порядковый индекс поверхности.

Библиотека:

surfacemanager

Описание:

Функция возвращает тип слоя, связанного с указанной поверхностью.

Примеры использования:

```
QApplication app(argc, argv);
gis_core_connection_t connection;
gis_core_request_parameters_t map;
gis_data_engine_context_t data_engine_ctx;
gis_render_sm_context_t sm_ctx;

gis_core_link_init( &connection );
gis_core_link_connect( &connection, 777 );
gis_core_databuffer_attach( &connection );
gis_core_request_parameters_init( &map );
gis_core_databuffer_data_request( &connection, &map, 1 );
gis_data_engine_alloc( 800, 600, GIS_DATA_ENGINE_MODE_SM, GIS_DATA_ENGINE_BPP_32,
↳ &connection, &data_engine_ctx );
gis_render_sm_alloc( &sm_ctx, data_engine_ctx, 1000, 0.5, 0 );

gis_render_sm_set_layer_type( sm_ctx, 0, GIS_RENDER_SM_LAYER_TYPE_USROBJ );
```

```

gis_render_sm_layer_type_t ltype = gis_render_sm_get_layer_type( sm_ctx, 0 );

if ( ltype != GIS_RENDER_SM_LAYER_TYPE_USROBJ )
{
    printf( "Incorrect layer type.\n" );
    return 1;
}

gis_render_sm_free( &sm_ctx );
gis_data_engine_free( &data_engine_ctx );
gis_core_databuffer_detach();
gis_core_link_destroy( &connection, true );

```

Классификация:

ПК ЦКИ для ЗОСРВ «Нейтрино»

Безопасность использования	
Обработчик прерываний	Нет
Обработчик сигналов	Нет
В потоке	Да

Тематические ссылки:

[gis_render_sm_context_t](#), [gis_render_sm_layer_type_t](#), [gis_render_sm_set_layer_type\(\)](#)

gis_render_sm_get_render_mode()

Получение режима рисования движка рендеринга Surface Manager.

Прототип:

```
#include <gis/gis_surfacemanager.h>

int gis_render_sm_get_render_mode( gis_render_sm_context_t sm_ctx );
```

Аргументы:

sm_ctx

Контекст движка рендеринга Surface Manager.

Библиотека:

surfacemanager

Описание:

Функция получает режим рисования движка рендеринга Surface Manager

Возвращаемое значение:

EINVAL

Некорректный контекст.

GIS_RENDER_SM_MODE_ASYNC

Асинхронный режим работы.

GIS_RENDER_SM_MODE_SYNC

Синхронный режим работы.

GIS_RENDER_SM_MODE_SEMI_SYNC

Полусинхронный режим, при котором область виджета рисуется в синхронном режиме, а фоновая область - в асинхронном.

Примеры использования:

```

QApplication app(argc, argv);
gis_core_connection_t connection;
gis_core_request_parameters_t map;
gis_data_engine_context_t data_engine_ctx;
gis_render_sm_context_t sm_ctx;

gis_core_link_init( &connection );
gis_core_link_connect( &connection, 777 );
gis_core_databuffer_attach( &connection );
gis_core_request_parameters_init( &map );
gis_core_databuffer_data_request( &connection, &map, 1 );
gis_data_engine_alloc( 800, 600, GIS_DATA_ENGINE_MODE_SM, GIS_DATA_ENGINE_BPP_32,
↪ &connection, &data_engine_ctx );
gis_render_sm_alloc( &sm_ctx, data_engine_ctx, 1000, 0.5, 0 );

gis_render_sm_set_render_mode( sm_ctx, GIS_RENDER_SM_MODE_ASYNC );

if ( gis_render_sm_get_render_mode( sm_ctx ) != GIS_RENDER_SM_MODE_ASYNC )
{
    printf( "Failed to get render mode\n" );
    return 1;
}

gis_render_sm_free( &sm_ctx );
gis_data_engine_free( &data_engine_ctx );
gis_core_databuffer_detach();
gis_core_link_destroy( &connection, true );

```

Классификация:

ПК ЦКИ для ЗОСРВ «Нейтрино»

Безопасность использования	
Обработчик прерываний	Нет
Обработчик сигналов	Нет
В потоке	Да

Поддерживается, начиная с ПК ЦКИ редакции 1.2

Тематические ссылки:

[gis_render_sm_context_t](#), [gis_render_sm_set_render_mode\(\)](#)

gis_render_sm_get_view_pixmap()

Получение поверхности окна отображения Surface Manager.

Прототип:

```
#include <gis/gis_surfacemanager.h>

void *gis_render_sm_get_view_pixmap( gis_render_sm_context_t sm_ctx, int32_point_t
↪ *viewport_position );
```

Аргументы:

sm_ctx

Контекст движка рендеринга Surface Manager.

viewport_position

Указатель на точку, в которую будет записано текущее смещение окна просмотра относительно поверхности.

Библиотека:

surfacemanager

Описание:

Функция возвращает указатель на поверхность отображения, а также текущее положение окна отображения относительно поверхности отображения.

Возвращаемое значение:

!NULL

Успешное завершение

NULL

Некорректные параметры функции

Примеры использования:

```

gis_render_sm_context_t      sm_ctx;
int32_point_t               viewport_pos;

void MapWidget::paintEvent(QPaintEvent * event)
{
    gis_render_sm_draw( sm_ctx );
    QPixmap pixmap = *(QPixmap *)gis_render_sm_get_view_pixmap( sm_ctx, &viewport_pos );
    QPainter painter( this );
    painter.drawPixmap( 0, 0, pixmap, viewport_pos.x, viewport_pos.y, width(), height()
↵ );
}

```

Классификация:

ПК ЦКИ для ЗОСРВ «Нейтрино»

Безопасность использования	
Обработчик прерываний	Нет
Обработчик сигналов	Нет
В потоке	Да

Тематические ссылки:

[gis_render_sm_context_t](#), [gis_render_sm_draw\(\)](#), [gis_render_sm_move\(\)](#), [gis_render_sm_rescale\(\)](#)

gis_render_sm_move()

Обработчик перемещения карты.

Прототип:

```
#include <gis/gis_surfacemanager.h>

int gis_render_sm_move( gis_render_sm_context_t sm_ctx, int dx, int dy );
```

Аргументы:

<i>sm_ctx</i>	Контекст движка рендеринга Surface Manager.
<i>dx</i>	Перемещение по горизонтали.
<i>dy</i>	Перемещение по вертикали.

Библиотека:

surfacemanager

Описание:

Функция смещает карту на заданное количество пикселей.

Возвращаемое значение:

<i>EOK</i>	Успешное завершение
<i>EINVAL</i>	Некорректные параметры функции
<i>ENOENT</i>	Контекст Surface Manager не был инициализирован
<i>EALREADY</i>	Величины перемещения равны нулю

Примеры использования:

```
gis_render_sm_context_t      sm_ctx;

void MapWidget::mouseMoveEvent( QMouseEvent *event )
{
    gis_render_sm_move( sm_ctx, event->pos().x(), event->pos().y() );
}
```

Классификация:

ПК ЦКИ для ЗОСРВ «Нейтрино»

Безопасность использования	
Обработчик прерываний	Нет
Обработчик сигналов	Нет
В потоке	Да

Тематические ссылки:

[gis_render_sm_context_t](#), [gis_render_sm_draw\(\)](#), [gis_render_sm_get_view_pixmap\(\)](#)

gis_render_sm_redraw_userobject()

Перерисовка пользовательского слоя.

Прототип:

```
#include <gis/gis_surfacemanager.h>

int gis_render_sm_redraw_userobject( gis_render_sm_context_t sm_ctx, bool redraw );
```

Аргументы:

sm_ctx

Контекст движка рендеринга Surface Manager.

redraw

Флаг, определяющий тип перерисовки. Варианты: 0 - обновить только пользовательскую поверхность окна отображения 1 - полная перерисовка пользовательского слоя

Библиотека:

surfacemanager

Описание:

Функция перерисовывает пользовательский слой.

Возвращаемое значение:

EOK

Успешное завершение

EINVAL

Некорректные параметры функции

Примеры использования:

```

vector<double_point_t>    degree_vector;
gis_userobject_t          poly;

uint32_t color = 0xFFFF00FF;
uint32_t border_color = 0xFFABABAB;

degree_vector.push_back( { 25, 20 } );
degree_vector.push_back( { 25, 25 } );

gis_render_sm_context_t    sm_ctx;
gis_data_engine_context_t  ctx;
gis_core_connection_t connection;
gis_core_link_init( &connection );

gis_core_link_connect( &connection, 777 );

uint32_t width = 100;
uint32_t height = 100;

gis_data_engine_alloc( width, height, GIS_DATA_ENGINE_MODE_SM, GIS_DATA_ENGINE_BPP_32,
↪ &connection, &ctx );
gis_render_sm_alloc( &sm_ctx, ctx, 1000, 0.5, 0 );

poly = gis_render_sm_userdata_add_polygon( sm_ctx, degree_vector.data(),
↪ degree_vector.size(), color, border_color );
int res = gis_render_sm_redraw_userobject( sm_ctx, true );

int retval = 0;

if ( res != EOK )
    retval += 1;

gis_render_sm_userdata_delete_object( sm_ctx, poly );
res = gis_render_sm_redraw_userobject( sm_ctx, true );

if ( res != EOK )
    retval += 1;

gis_render_sm_free( &sm_ctx );
gis_data_engine_free( &ctx );
gis_core_databuffer_detach();
gis_core_link_destroy( &connection, true );

return retval;

```

Классификация:

ПК ЦКИ для ЗОСРВ «Нейтрино»

Безопасность использования	
<i>Обработчик прерываний</i>	Нет
<i>Обработчик сигналов</i>	Нет
<i>В потоке</i>	Да

Тематические ссылки:

`gis_render_sm_context_t`, `gis_render_sm_userdata_add_polygon()`

gis_render_sm_rescale()

Масштабирование карты.

Прототип:

```
#include <gis/gis_surfacemanager.h>

int gis_render_sm_rescale( gis_render_sm_context_t sm_ctx, uint32_t new_scale_denom );
```

Аргументы:

sm_ctx
Контекст движка рендеринга Surface Manager.

new_scale_denom
Новый знаменатель масштаба.

Библиотека:

surfacemanager

Описание:

Функция устанавливает новый масштаб отображения 1:new_scale_denom.

Возвращаемое значение:

EOK
Успешное завершение

EINVAL
Некорректные параметры функции

ENOENT
Контекст Surface Manager не был инициализирован

EALREADY
Передан текущий масштаб отображения

Примеры использования:

```
gis_render_sm_context_t    sm_ctx;
uint32_t real_scale_denom = 10000;

if ( gis_render_sm_rescale( sm_ctx, real_scale_denom ) != EOK ) {
    printf( "gis_render_sm_rescale failed.\n" );
    return -1;
}
```

Классификация:

ПК ЦКИ для ЗОСРВ «Нейтрино»

Безопасность использования	
Обработчик прерываний	Нет
Обработчик сигналов	Нет
В потоке	Да

Тематические ссылки:

[gis_render_sm_context_t](#), [gis_render_sm_draw\(\)](#), [gis_render_sm_move\(\)](#)

gis_render_sm_set_layer_data_sources()

Назначение форматов карт для отображения в заданной поверхности.

Прототип:

```
#include <gis/gis_surfacemanager.h>

int gis_render_sm_set_layer_data_sources( gis_render_sm_context_t    sm_ctx,
                                          gis_render_sm_layer_type_t  layer_type,
                                          uint32_t                    ↵
                                          number_of_data_sources,
                                          gis_core_map_data_source_t    *data_sources );
```

Аргументы:

sm_ctx

Контекст движка рендеринга Surface Manager.

layer_type

Тип слоя `gis_render_sm_layer_type_t`.

number_of_data_sources

Размер массива форматов карт `data_sources`.

data_sources

Массив из значений `gis_core_map_data_source_t`, которые определяют, какие форматы карт требуется отобразить в поверхности `z_idx`.

Библиотека:

surfacemanager

Описание:

Функция устанавливает требуемые форматы карт для отображения в заданной поверхности.

Возвращаемое значение:

EOK

Успешное завершение

EINVAL

Некорректные параметры функции

ENOENT

Контекст Surface Manager не был инициализирован

EFAULT

Произведена попытка поместить недопустимые форматы в слой. Например, векторные форматы в GIS_RENDER_SM_LAYER_TYPE_MTW.



Данная функция не производит никаких действий над слоем GIS_RENDER_SM_LAYER_TYPE_U↔SROBJ.

Примеры использования:

```
QApplication app(argc, argv);
gis_core_connection_t connection;
gis_core_request_parameters_t map;
gis_data_engine_context_t data_engine_ctx;
gis_render_sm_context_t sm_ctx;

gis_core_link_init( &connection );
gis_core_link_connect( &connection, 777 );
gis_core_databuffer_attach( &connection );
gis_core_request_parameters_init( &map );
gis_core_databuffer_data_request( &connection, &map, 1 );
gis_data_engine_alloc( 800, 600, GIS_DATA_ENGINE_MODE_SM, GIS_DATA_ENGINE_BPP_32,
↪ &connection, &data_engine_ctx );
gis_render_sm_alloc( &sm_ctx, data_engine_ctx, 1000, 0.5, 0 );

gis_render_sm_set_layer_type( sm_ctx, 0, GIS_RENDER_SM_LAYER_TYPE_MTW );
std::vector<gis_core_map_data_source_t> ds;
ds.reserve( 2 );
ds.push_back( GIS_CORE_MAP_DATA_SOURCE_MTW );
ds.push_back( GIS_CORE_MAP_DATA_SOURCE_GEOTIFF );

gis_render_sm_set_layer_data_sources( sm_ctx, GIS_RENDER_SM_LAYER_TYPE_MTW, ds.size(),
↪ &ds[0] );

gis_render_sm_free( &sm_ctx );
gis_data_engine_free( &data_engine_ctx );
gis_core_databuffer_detach();
gis_core_link_destroy( &connection, true );
```

Поддерживается, начиная с ПК ЦКИ редакции 1.1.0

Поддерживается, начиная с ПК ЦКИ редакции 1.2

Классификация:

Безопасность использования	
Обработчик прерываний	Нет
Обработчик сигналов	Нет
В потоке	Да

Тематические ссылки:

`gis_render_sm_context_t`, `gis_render_sm_layer_type_t`, `gis_render_sm_set_layer_type()`

gis_render_sm_set_layer_type()

Назначение типа слоя заданной поверхности.

Прототип:

```
#include <gis/gis_surfacemanager.h>

int gis_render_sm_set_layer_type( gis_render_sm_context_t sm_ctx, uint8_t z_idx,
↪ gis_render_sm_layer_type_t layer_type );
```

Аргументы:

sm_ctx

Контекст движка рендеринга Surface Manager.

z_idx

Индекс поверхности (наибольший индекс соответствует верхней поверхности).

layer_type

Тип слоя `gis_render_sm_layer_type_t`.

Библиотека:

surfacemanager

Описание:

Функция устанавливает соответствие конкретного слоя заданной поверхности.

Возвращаемое значение:

EOK

Успешное завершение

EINVAL

Некорректные параметры функции

ENOENT

Контекст Surface Manager не был инициализирован

EALREADY

Слой уже назначен этой поверхности

Примеры использования:

```

QApplication app(argc, argv);
gis_core_connection_t connection;
gis_core_request_parameters_t map;
gis_data_engine_context_t data_engine_ctx;
gis_render_sm_context_t sm_ctx;

gis_core_link_init( &connection );
gis_core_link_connect( &connection, 777 );
gis_core_databuffer_attach( &connection );
gis_core_request_parameters_init( &map );
gis_core_databuffer_data_request( &connection, &map, 1 );
gis_data_engine_alloc( 800, 600, GIS_DATA_ENGINE_MODE_SM, GIS_DATA_ENGINE_BPP_32,
↪ &connection, &data_engine_ctx );
gis_render_sm_alloc( &sm_ctx, data_engine_ctx, 1000, 0.5, 0 );

gis_render_sm_set_layer_type( sm_ctx, 0, GIS_RENDER_SM_LAYER_TYPE_USROBJ );

gis_render_sm_layer_type_t ltype = gis_render_sm_get_layer_type( sm_ctx, 0 );

if ( ltype != GIS_RENDER_SM_LAYER_TYPE_USROBJ )
{
    printf( "Incorrect layer type.\n" );
    return 1;
}

gis_render_sm_free( &sm_ctx );
gis_data_engine_free( &data_engine_ctx );
gis_core_databuffer_detach();
gis_core_link_destroy( &connection, true );

```

Классификация:

ПК ЦКИ для ЗОСРВ «Нейтрино»

Безопасность использования	
Обработчик прерываний	Нет
Обработчик сигналов	Нет
В потоке	Да

Тематические ссылки:

[gis_render_sm_context_t](#), [gis_render_sm_layer_type_t](#), [gis_render_sm_move\(\)](#),

gis_render_sm_set_render_mode()

Установка режима рисования движка рендеринга Surface Manager.

Прототип:

```
#include <gis/gis_surfacemanager.h>

int gis_render_sm_set_render_mode( gis_render_sm_context_t sm_ctx, uint8_t render_mode );
```

Аргументы:

sm_ctx

Контекст движка рендеринга Surface Manager.

render_mode

Режим рисования. Возможные варианты: GIS_RENDER_SM_MODE_ASYNC - асинхронный режим работы. GIS_RENDER_SM_MODE_SYNC - синхронный режим работы. GIS_RENDER_SM_MODE_SEMI_SYNC - полусинхронный режим, при котором область виджета рисуется в синхронном режиме, а фоновая область - в асинхронном.

Библиотека:

surfacemanager

Описание:

Функция устанавливает режим рисования движка рендеринга Surface Manager

Возвращаемое значение:

EOK

Успешное завершение

EINVAL

Некорректные параметры функции

ENOENT

Контекст Surface Manager не был инициализирован

EFAULT

Передан некорректный режим рисования

Примеры использования:

```

QApplication app(argc, argv);
gis_core_connection_t connection;
gis_core_request_parameters_t map;
gis_data_engine_context_t data_engine_ctx;
gis_render_sm_context_t sm_ctx;

gis_core_link_init( &connection );
gis_core_link_connect( &connection, 777 );
gis_core_databuffer_attach( &connection );
gis_core_request_parameters_init( &map );
gis_core_databuffer_data_request( &connection, &map, 1 );
gis_data_engine_alloc( 800, 600, GIS_DATA_ENGINE_MODE_SM, GIS_DATA_ENGINE_BPP_32,
↪ &connection, &data_engine_ctx );
gis_render_sm_alloc( &sm_ctx, data_engine_ctx, 1000, 0.5, 0 );

gis_render_sm_set_render_mode( sm_ctx, GIS_RENDER_SM_MODE_ASYNC );

if ( gis_render_sm_get_render_mode( sm_ctx ) != GIS_RENDER_SM_MODE_ASYNC )
{
    printf( "Failed to set render mode\n" );
    return 1;
}

gis_render_sm_free( &sm_ctx );
gis_data_engine_free( &data_engine_ctx );
gis_core_databuffer_detach();
gis_core_link_destroy( &connection, true );

```

Классификация:

ПК ЦКИ для ЗОСРВ «Нейтрино»

Безопасность использования	
Обработчик прерываний	Нет
Обработчик сигналов	Нет
В потоке	Да

Тематические ссылки:

[gis_render_sm_context_t](#), [gis_render_sm_get_render_mode\(\)](#), [gis_render_sm_draw\(\)](#)

gis_render_sm_set_update_func()

Установка функции оповещения о состоянии рисования Surface Manager.

Прототип:

```
#include <gis/gis_surfacemanager.h>

void gis_render_sm_set_update_func( gis_render_sm_context_t sm_ctx, SMCallbacks
↳ *update_class );
```

Аргументы:

sm_ctx

Контекст движка рендеринга Surface Manager.

update_class

Указатель на объект класса `class SMCallbacks`.

Библиотека:

surfacemanager

Описание:

Функция устанавливает обработчик оповещения об изменении состояния рисования Surface Manager.

Примеры использования:

```
MapWidget w; // inherited from SMCallbacks class

QApplication app(argc, argv);
gis_core_connection_t connection;
gis_core_request_parameters_t map;
gis_data_engine_context_t data_engine_ctx;
gis_render_sm_context_t sm_ctx;

gis_core_link_init( &connection );
gis_core_link_connect( &connection, 777 );
gis_core_databuffer_attach( &connection );
gis_core_request_parameters_init( &map );
gis_core_databuffer_data_request( &connection, &map, 1 );
gis_data_engine_alloc( 800, 600, GIS_DATA_ENGINE_MODE_SM, GIS_DATA_ENGINE_BPP_32,
↳ &connection, &data_engine_ctx );
gis_render_sm_alloc( &sm_ctx, data_engine_ctx, 1000, 0.5, 0 );
```

```

gis_render_sm_set_update_func( sm_ctx, &w );

gis_render_sm_draw( sm_ctx );

if ( surface_completed != 2 )
{
    printf( "Failed to set update func\n" );
    return 1;
}

gis_render_sm_free( &sm_ctx );
gis_data_engine_free( &data_engine_ctx );
gis_core_databuffer_detach();
gis_core_link_destroy( &connection, true );

```

Классификация:

ПК ЦКИ для ЗОСРВ «Нейтрино»

Безопасность использования	
Обработчик прерываний	Нет
Обработчик сигналов	Нет
В потоке	Да

Тематические ссылки:

[gis_render_sm_context_t](#), [gis_render_sm_draw\(\)](#), [gis_render_sm_move\(\)](#), [gis_render_sm_rescale\(\)](#)

gis_render_sm_sync()

Задать тип следующей операции рисования.

Прототип:

```
#include <gis/gis_surfacemanager.h>

int gis_render_sm_sync( gis_render_sm_context_t sm_ctx, bool redraw );
```

Аргументы:

sm_ctx

Контекст движка рендеринга Surface Manager.

redraw

Флаг, определяющий тип перерисовки. Варианты: 0 - обновить только поверхность окна отображения 1 - полная перерисовка всех слоёв

Библиотека:

surfacemanager

Описание:

Функция принудительно устанавливает тип следующей операции рисования Surface Manager при вызове `gis_render_sm_draw()`.

Возвращаемое значение:

EOK

Успешное завершение

EINVAL

Некорректные параметры функции

ENOENT

Контекст Surface Manager не был инициализирован

Примеры использования:

```

QApplication app(argc, argv);
gis_core_connection_t connection;
gis_core_request_parameters_t map;
gis_data_engine_context_t data_engine_ctx;
gis_render_sm_context_t sm_ctx;

gis_core_link_init( &connection );
gis_core_link_connect( &connection, 777 );
gis_core_databuffer_attach( &connection );
gis_core_request_parameters_init( &map );
gis_core_databuffer_data_request( &connection, &map, 1 );
gis_data_engine_alloc( 800, 600, GIS_DATA_ENGINE_MODE_SM, GIS_DATA_ENGINE_BPP_32,
↪ &connection, &data_engine_ctx );
gis_render_sm_alloc( &sm_ctx, data_engine_ctx, 1000, 0.5, 0 );

if ( gis_render_sm_sync( sm_ctx, true ) != EOK )
{
    printf( "gis_render_sm_sync() failed\n" );
    return 1;
}

gis_render_sm_draw( sm_ctx );

gis_render_sm_free( &sm_ctx );
gis_data_engine_free( &data_engine_ctx );
gis_core_databuffer_detach();
gis_core_link_destroy( &connection, true );

```

Классификация:

ПК ЦКИ для ЗОСРВ «Нейтрино»

Безопасность использования	
Обработчик прерываний	Нет
Обработчик сигналов	Нет
В потоке	Да

Тематические ссылки:

[gis_render_sm_context_t](#), [gis_render_sm_draw\(\)](#), [gis_render_sm_move\(\)](#), [gis_render_sm_rescale\(\)](#)

gis_render_sm_update()

Обновление параметров Surface Manager.

Прототип:

```
#include <gis/gis_surfacemanager.h>

int gis_render_sm_update( gis_render_sm_context_t sm_ctx, uint32_t surface_extension_px,
↳ double blit_region_rate );
```

Аргументы:

sm_ctx

Контекст движка рендеринга Surface Manager.

new_scale_denom

Новый знаменатель масштаба.

Библиотека:

surfacemanager

Описание:

Функция устанавливает новые параметры Surface Manager: *offscreen_ext_px* и *blit_region_rate*. В процессе выполнения поверхность отображения будет повторно выделена. Если изменяются размеры окна отображения функцией *gis_data_engine_set_canvas_size()*, то обязательно необходимо затем вызвать текущую функцию.

Возвращаемое значение:

EOK

Успешное завершение

EINVAL

Некорректные параметры функции

ENOENT

Контекст Surface Manager не был инициализирован

ENOMEM

Не удалось выделить поверхность

EFAULT

Ошибка выполнения

Примеры использования:

```

gis_render_sm_context_t    sm_ctx;
gis_data_engine_context_t  ctx;

void MapWidget::resizeEvent(QResizeEvent *event)
{
    gis_data_engine_set_canvas_size( ctx, event->size().width(), event->size().height()
↵ );

    if ( gis_render_sm_update( sm_ctx, offscreen_ext_px, 0.5 ) != EOK ) {
        printf( "gis_render_sm_update failed - Need to rollback.\n" );
    }
}

```

Классификация:

ПК ЦКИ для ЗОСРВ «Нейтрино»

Безопасность использования	
Обработчик прерываний	Нет
Обработчик сигналов	Нет
В потоке	Да

Тематические ссылки:

[gis_render_sm_context_t](#), [gis_render_sm_draw\(\)](#), [gis_render_sm_move\(\)](#), [gis_render_sm_rescale\(\)](#)

gis_render_sm_userdata_add_bitmap()

Добавление изображения.

Прототип:

```
#include <gis/gis_surfacemanager.h>

gis_userobject_t gis_render_sm_userdata_add_bitmap( gis_render_sm_context_t sm_ctx,
↪ double_point_t *array, int length, void *image );
```

Аргументы:

<i>sm_ctx</i>	Контекст движка рендеринга Surface Manager.
<i>array</i>	Градусная точка привязки.
<i>length</i>	Количество точек.
<i>image</i>	Указатель на изображение.

Библиотека:

surfacemanager

Описание:

Функция создаёт пользовательский объект типа изображение.

Возвращаемое значение:

<i>!NULL</i>	Успешное завершение
<i>NULL</i>	Некорректные параметры функции

Примеры использования:

```

vector<double_point_t>      degree_vector;
gis_userobject_t            img;
QImage                      image;

degree_vector.push_back( { 25, 20 } );

gis_render_sm_context_t      sm_ctx;
gis_data_engine_context_t    ctx;
gis_core_connection_t        connection;
gis_core_link_init( &connection );

gis_core_link_connect( &connection, 777 );

uint32_t width = 100;
uint32_t height = 100;

gis_data_engine_alloc( width, height, GIS_DATA_ENGINE_MODE_SM, GIS_DATA_ENGINE_BPP_32,
↪ &connection, &ctx );
gis_render_sm_alloc( &sm_ctx, ctx, 1000, 0.5, 0 );

img = gis_render_sm_userdata_add_bitmap( sm_ctx,
↪ degree_vector.data(), degree_vector.size(), (void*)&image );

int retval = 0;

if ( !img )
    retval += 1;

gis_render_sm_free( &sm_ctx );
gis_data_engine_free( &ctx );
gis_core_databuffer_detach();
gis_core_link_destroy( &connection, true );

return retval;

```

Классификация:

ПК ЦКИ для ЗОСРВ «Нейтрино»

Безопасность использования	
Обработчик прерываний	Нет
Обработчик сигналов	Нет
В потоке	Да

Тематические ссылки:

[gis_render_sm_context_t](#), [gis_render_sm_userdata_add_polygon\(\)](#)

gis_render_sm_userdata_add_contour_object()

Добавление точки контура.

Прототип:

```
#include <gis/gis_surfacemanager.h>

gis_userobject_t gis_render_sm_userdata_add_contour_object( gis_render_sm_context_t sm_ctx,
↪ double_point_t *array, int length, uint32_t color, int pen_width );
```

Аргументы:

<i>sm_ctx</i>	Контекст движка рендеринга Surface Manager.
<i>array</i>	Массив градусных точек.
<i>length</i>	Количество точек.
<i>color</i>	Цвет объекта.
<i>pen_width</i>	Толщина линии.

Библиотека:

surfacemanager

Описание:

Функция создаёт пользовательский объект типа точка контура. Используется для построения контуров других пользовательских объектов.

Возвращаемое значение:

<i>!NULL</i>	Успешное завершение
<i>NULL</i>	Некорректные параметры функции

Примеры использования:

```

QApplication app(argc, argv);
vector<double_point_t> degree_vector;
gis_userobject_t contour_point;

degree_vector.push_back( { 25, 20 } );

uint32_t color = 0xFFFF00FF;
uint32_t border_color = 0xFFABABAB;

gis_render_sm_context_t sm_ctx;
gis_data_engine_context_t ctx;
gis_core_connection_t connection;
gis_core_link_init( &connection );

gis_core_link_connect( &connection, 777 );

uint32_t width = 100;
uint32_t height = 100;

gis_data_engine_alloc( width, height, GIS_DATA_ENGINE_MODE_SM, GIS_DATA_ENGINE_BPP_32,
↪ &connection, &ctx );
gis_render_sm_alloc( &sm_ctx, ctx, 1000, 0.5, 0 );

contour_point = gis_render_sm_userdata_add_contour_object( sm_ctx, degree_vector.data(),
↪ degree_vector.size(), color, border_color );

int retval = 0;

if ( !contour_point )
    retval += 1;

gis_render_sm_free( &sm_ctx );
gis_data_engine_free( &ctx );
gis_core_databuffer_detach();
gis_core_link_destroy( &connection, true );

return retval;

```

Классификация:

ПК ЦКИ для ЗОСРВ «Нейтрино»

Безопасность использования	
Обработчик прерываний	Нет
Обработчик сигналов	Нет
В потоке	Да

Поддерживается, начиная с ПК ЦКИ редакции 1.4

Тематические ссылки:

`gis_render_sm_context_t, gis_render_sm_userdata_delete_contour_object()`

gis_render_sm_userdata_add_polygon()

Добавление полигона.

Прототип:

```
#include <gis/gis_surfacemanager.h>

gis_userobject_t gis_render_sm_userdata_add_polygon( gis_render_sm_context_t sm_ctx,
↪ double_point_t *array, int length, uint32_t color, uint32_t border_color );
```

Аргументы:

<i>sm_ctx</i>	Контекст движка рендеринга Surface Manager.
<i>array</i>	Массив градусных точек.
<i>length</i>	Количество точек.
<i>color</i>	Цвет заливки.
<i>border_color</i>	Цвет границы полигона.

Библиотека:

surfacemanager

Описание:

Функция создаёт пользовательский объект типа полигон.

Возвращаемое значение:

<i>!NULL</i>	Успешное завершение
<i>NULL</i>	Некорректные параметры функции

Примеры использования:

```

vector<double_point_t>    degree_vector;
gis_userobject_t          poly;

uint32_t color = 0xFFFF00FF;
uint32_t border_color = 0xFFABABAB;

degree_vector.push_back( { 25, 20 } );
degree_vector.push_back( { 25, 25 } );

gis_render_sm_context_t    sm_ctx;
gis_data_engine_context_t  ctx;
gis_core_connection_t connection;
gis_core_link_init( &connection );

gis_core_link_connect( &connection, 777 );

uint32_t width = 100;
uint32_t height = 100;

gis_data_engine_alloc( width, height, GIS_DATA_ENGINE_MODE_SM, GIS_DATA_ENGINE_BPP_32,
↪ &connection, &ctx );
gis_render_sm_alloc( &sm_ctx, ctx, 1000, 0.5, 0 );

poly = gis_render_sm_userdata_add_polygon( sm_ctx,
↪ degree_vector.data(), degree_vector.size(), color, border_color );

int retval = 0;

if ( !poly )
    retval += 1;

gis_render_sm_free( &sm_ctx );
gis_data_engine_free( &ctx );
gis_core_databuffer_detach();
gis_core_link_destroy( &connection, true );

return retval;

```

Классификация:

ПК ЦКИ для ЗОСРВ «Нейтрино»

Безопасность использования	
Обработчик прерываний	Нет
Обработчик сигналов	Нет
В потоке	Да

Тематические ссылки:

`gis_render_sm_context_t`, `gis_render_sm_calculate_polygon()`

gis_render_sm_userdata_add_polyline()

Добавление полилинии.

Прототип:

```
#include <gis/gis_surfacemanager.h>

gis_userobject_t gis_render_sm_userdata_add_polyline( gis_render_sm_context_t sm_ctx,
↪ double_point_t *array, int length, uint32_t color, int pen_width, int line_type );
```

Аргументы:

sm_ctx

Контекст движка рендеринга Surface Manager.

array

Массив градусных точек.

length

Количество точек.

color

Цвет линии.

pen_width

Толщина линии.

line_type

Тип линии. Возможные варианты: 0 - сплошная линия, 1 - пунктирная линия.

Библиотека:

surfacemanager

Описание:

Функция создаёт пользовательский объект типа полилиния.

Возвращаемое значение:

!NULL

Успешное завершение

NULL

Некорректные параметры функции

Примеры использования:

```
vector<double_point_t>    degree_vector;
gis_userobject_t          line;

uint32_t color = 0xFFAFAFAF;
int pen_width = 10;
int line_type = 1;

degree_vector.push_back( { 25, 20 } );
degree_vector.push_back( { 25, 25 } );

gis_render_sm_context_t    sm_ctx;
gis_data_engine_context_t  ctx;
gis_core_connection_t connection;
gis_core_link_init( &connection );

gis_core_link_connect( &connection, 777 );

uint32_t width = 100;
uint32_t height = 100;

gis_data_engine_alloc( width, height, GIS_DATA_ENGINE_MODE_SM, GIS_DATA_ENGINE_BPP_32,
↪ &connection, &ctx );
gis_render_sm_alloc( &sm_ctx, ctx, 1000, 0.5, 0 );

line = gis_render_sm_userdata_add_polyline( sm_ctx,
↪ degree_vector.data(), degree_vector.size(), color, pen_width, line_type );

int retval = 0;

if ( !line )
    retval += 1;

gis_render_sm_free( &sm_ctx );
gis_data_engine_free( &ctx );
gis_core_databuffer_detach();
gis_core_link_destroy( &connection, true );

return retval;
```

Классификация:

ПК ЦКИ для ЗОСРВ «Нейтрино»

Безопасность использования	
Обработчик прерываний	Нет
Обработчик сигналов	Нет
В потоке	Да

Тематические ссылки:

`gis_render_sm_context_t, gis_render_sm_calculate_distance()`

gis_render_sm_userdata_delete_contour_object()

Удаление точки контура.

Прототип:

```
#include <gis/gis_surfacemanager.h>

int gis_render_sm_userdata_delete_contour_object( gis_render_sm_context_t sm_ctx,
↪ gis_userobject_t obj );
```

Аргументы:

sm_ctx

Контекст движка рендеринга Surface Manager.

obj

Указатель на пользовательский объект.

Библиотека:

surfacemanager

Описание:

Функция удаляет пользовательский объект типа точка контура.

Возвращаемое значение:

EOK

Успешное завершение

EINVAL

Некорректные параметры функции

Примеры использования:

```

QApplication app(argc, argv);
vector<double_point_t> degree_vector;
gis_userobject_t contour_point;

degree_vector.push_back( { 25, 20 } );

uint32_t color = 0xFFFF00FF;
uint32_t border_color = 0xFFABABAB;

gis_render_sm_context_t sm_ctx;
gis_data_engine_context_t ctx;
gis_core_connection_t connection;
gis_core_link_init( &connection );

gis_core_link_connect( &connection, 777 );

uint32_t width = 100;
uint32_t height = 100;

gis_data_engine_alloc( width, height, GIS_DATA_ENGINE_MODE_SM, GIS_DATA_ENGINE_BPP_32,
↪ &connection, &ctx );
gis_render_sm_alloc( &sm_ctx, ctx, 1000, 0.5, 0 );

contour_point = gis_render_sm_userdata_add_contour_object( sm_ctx, degree_vector.data(),
↪ degree_vector.size(), color, border_color );

gis_render_sm_userdata_delete_contour_object( sm_ctx, contour_point );

int retval = 0;

if ( contour_point )
    retval += 1;

gis_render_sm_free( &sm_ctx );
gis_data_engine_free( &ctx );
gis_core_databuffer_detach();
gis_core_link_destroy( &connection, true );

return retval;

```

Классификация:

ПК ЦКИ для ЗОСРВ «Нейтрино»

Безопасность использования	
Обработчик прерываний	Нет
Обработчик сигналов	Нет
В потоке	Да

Поддерживается, начиная с ПК ЦКИ редакции 1.4

Тематические ссылки:

`gis_render_sm_context_t, gis_render_sm_userdata_add_contour_object()`

gis_render_sm_userdata_delete_object()

Удаление пользовательского объекта.

Прототип:

```
#include <gis/gis_surfacemanager.h>

int gis_render_sm_userdata_delete_object( gis_render_sm_context_t sm_ctx, gis_userobject_t
↵ obj );
```

Аргументы:

<i>sm_ctx</i>	Контекст движка рендеринга Surface Manager.
<i>obj</i>	Указатель на пользовательский объект.

Библиотека:

surfacemanager

Описание:

Функция удаляет пользовательский объект.

Возвращаемое значение:

<i>EOK</i>	Успешное завершение
<i>EINVAL</i>	Некорректные параметры функции

Примеры использования:

```

vector<double_point_t>    degree_vector;
gis_userobject_t          poly;

uint32_t color = 0xFFFF00FF;
uint32_t border_color = 0xFFABABAB;

degree_vector.push_back( { 25, 20 } );
degree_vector.push_back( { 25, 25 } );

gis_render_sm_context_t    sm_ctx;
gis_data_engine_context_t  ctx;
gis_core_connection_t connection;
gis_core_link_init( &connection );

gis_core_link_connect( &connection, 777 );

uint32_t width = 100;
uint32_t height = 100;

gis_data_engine_alloc( width, height, GIS_DATA_ENGINE_MODE_SM, GIS_DATA_ENGINE_BPP_32,
↪ &connection, &ctx );
gis_render_sm_alloc( &sm_ctx, ctx, 1000, 0.5, 0 );

poly = gis_render_sm_userdata_add_polygon( sm_ctx,
↪ degree_vector.data(), degree_vector.size(), color, border_color );

int res = gis_render_sm_userdata_delete_object( sm_ctx, poly );

int retval = 0;

if ( res != EOK )
    retval += 1;

gis_render_sm_free( &sm_ctx );
gis_data_engine_free( &ctx );
gis_core_databuffer_detach();
gis_core_link_destroy( &connection, true );

return retval;

```

Классификация:

ПК ЦКИ для ЗОСРВ «Нейтрино»

Безопасность использования	
Обработчик прерываний	Нет
Обработчик сигналов	Нет
В потоке	Да

Тематические ссылки:

`gis_render_sm_context_t`, `gis_render_sm_userdata_add_polygon()`

gis_render_sm_userdata_edit_object_color()

Изменение цвета объекта.

Прототип:

```
#include <gis/gis_surfacemanager.h>

int gis_render_sm_userdata_edit_object_color( gis_render_sm_context_t sm_ctx,
↪ gis_userobject_t obj, uint32_t color, uint32_t border_color );
```

Аргументы:

<i>sm_ctx</i>	Контекст движка рендеринга Surface Manager.
<i>obj</i>	Указатель на объект.
<i>color</i>	Новый основной цвет.
<i>border_color</i>	Новый цвет границы (применимо к полигону).

Библиотека:

surfacemanager

Описание:

Функция изменяет цвет пользовательского объекта (неприменимо к изображениям).

Возвращаемое значение:

<i>EOK</i>	Успешное завершение
<i>EINVAL</i>	Некорректные параметры функции

Примеры использования:

```

vector<double_point_t>      degree_vector;
gis_userobject_t            line;

degree_vector.push_back( { 25, 20 } );
degree_vector.push_back( { 25, 25 } );

gis_render_sm_context_t      sm_ctx;
gis_data_engine_context_t    ctx;
gis_core_connection_t        connection;
gis_core_link_init( &connection );

gis_core_link_connect( &connection, 777 );

uint32_t width = 100;
uint32_t height = 100;

gis_data_engine_alloc( width, height, GIS_DATA_ENGINE_MODE_SM, GIS_DATA_ENGINE_BPP_32,
↪ &connection, &ctx );
gis_render_sm_alloc( &sm_ctx, ctx, 1000, 0.5, 0 );

line = gis_render_sm_userdata_add_polyline( sm_ctx,
↪ degree_vector.data(), degree_vector.size(), 0xFF00FF00, 10, 1 );

int res = gis_render_sm_userdata_edit_object_color( sm_ctx, line, 0xFFFF0000, 0 );

int retval = 0;

if ( res != EOK )
    retval += 1;

gis_render_sm_free( &sm_ctx );
gis_data_engine_free( &ctx );
gis_core_databuffer_detach();
gis_core_link_destroy( &connection, true );

return retval;

```

Классификация:

ПК ЦКИ для ЗОСРВ «Нейтрино»

Безопасность использования	
Обработчик прерываний	Нет
Обработчик сигналов	Нет
В потоке	Да

Тематические ссылки:

```
gis_render_sm_context_t, gis_render_sm_userdata_add_polyline()
```

gis_render_sm_userdata_edit_object_points()

Изменение точек объекта.

Прототип:

```
#include <gis/gis_surfacemanager.h>

int gis_render_sm_userdata_edit_object_points( gis_render_sm_context_t sm_ctx,
↪ gis_userobject_t obj, double_point_t* array, int length );
```

Аргументы:

<i>sm_ctx</i>	Контекст движка рендеринга Surface Manager.
<i>obj</i>	Указатель на объект.
<i>array</i>	Новый массив точек.
<i>length</i>	Длина массива.

Библиотека:

surfacemanager

Описание:

Функция изменяет точки пользовательского объекта.

Возвращаемое значение:

<i>EOK</i>	Успешное завершение
<i>EINVAL</i>	Некорректные параметры функции

Примеры использования:

```

vector<double_point_t>    degree_vector;
gis_userobject_t          line;

degree_vector.push_back( { 25, 20 } );
degree_vector.push_back( { 25, 25 } );

gis_render_sm_context_t    sm_ctx;
gis_data_engine_context_t  ctx;
gis_core_connection_t      connection;
gis_core_link_init( &connection );

gis_core_link_connect( &connection, 777 );

uint32_t width = 100;
uint32_t height = 100;

gis_data_engine_alloc( width, height, GIS_DATA_ENGINE_MODE_SM, GIS_DATA_ENGINE_BPP_32,
↪ &connection, &ctx );
gis_render_sm_alloc( &sm_ctx, ctx, 1000, 0.5, 0 );

line = gis_render_sm_userdata_add_polyline( sm_ctx,
↪ degree_vector.data(), degree_vector.size(), 0xFF00FF00, 10, 1 );

degree_vector.push_back( { 25, 30 } );

int res = gis_render_sm_userdata_edit_object_points( sm_ctx, line,
↪ degree_vector.data(), degree_vector.size() );

int retval = 0;

if ( res != EOK )
    retval += 1;

gis_render_sm_free( &sm_ctx );
gis_data_engine_free( &ctx );
gis_core_databuffer_detach();
gis_core_link_destroy( &connection, true );

return retval;

```

Классификация:

ПК ЦКИ для ЗОСРВ «Нейтрино»

Безопасность использования	
Обработчик прерываний	Нет
Обработчик сигналов	Нет
В потоке	Да

Тематические ссылки:

`gis_render_sm_context_t, gis_render_sm_userdata_edit_object_color()`

gis_render_sm_context_t

Контекст движка рендеринга Surface Manager.

Формат:

```
#include <gis/gis_surfacemanager.h>

typedef void *      gis_render_sm_context_t;
```

Описание:

Данный контекст обеспечивает управление параметрами движка рендеринга Surface Manager.

Классификация:

ПК ЦКИ для ЗОСРВ «Нейтрино»

gis_render_sm_layer_type_t

Слой движка рендеринга.

Формат:

```
#include <gis/gis_surfacemanager.h>

typedef enum {
    GIS_RENDER_SM_LAYER_TYPE_DISABLED = 0,
    GIS_RENDER_SM_LAYER_TYPE_2DMAP,
    GIS_RENDER_SM_LAYER_TYPE_USROBJ,
    GIS_RENDER_SM_LAYER_TYPE_MTW
} gis_render_sm_layer_type_t;
```

Описание:

Перечисление `gis_render_sm_layer_type_t` включает следующие значения:

- `GIS_RENDER_SM_LAYER_TYPE_DISABLED` — слой выключен, его поверхность исключена из рисования.
- `GIS_RENDER_SM_LAYER_TYPE_2DMAP` — слой рисования векторных данных.
- `GIS_RENDER_SM_LAYER_TYPE_USROBJ` — слой рисования пользовательских объектов.
- `GIS_RENDER_SM_LAYER_TYPE_MTW` — слой рисования растровых данных.

Классификация:

ПК ЦКИ для ЗОСРВ «Нейтрино»

Тематические ссылки:

`gis_render_sm_set_layer_type()`

gis_userobject_t

Тип пользовательского объекта движка рендеринга *Surface Manager*.

Формат:

```
#include <gis/gis_surfacemanager.h>

typedef void *gis_userobject_t;
```

Описание:

Данный тип предназначен для работы с пользовательскими объектами движка рендеринга *Surface Manager*.

Поддерживается, начиная с ПК ЦКИ редакции 1.3



Данный тип до релиза ПК ЦКИ версии 1.3 именовался `userobject_t`.

Классификация:

ПК ЦКИ для ЗОСРВ «Нейтрино»

gis_render_rb_alloc()

Создание контекста движка рендеринга.

Прототип:

```
#include <gis/gis_renderbuffer.h>

int gis_render_rb_alloc( gis_render_rb_context_t *rb_ctx, gis_data_engine_context_t
↪ data_engine_ctx, QPainter *painter);
```

Аргументы:

rb_ctx

Указатель на контекст движка рендеринга Render Buffer.

data_engine_ctx

Указатель на контекст движка рендеринга Data Engine.

painter

Указатель на QPainter

Библиотека:

renderbuffer

Описание:

Функция создаёт контекст движка рендеринга Render Buffer и инициализирует его переданными параметрами.

Возвращаемое значение:

EOK

Успешное завершение

EINVAL

Некорректные параметры функции

ENOMEM

Недостаточно памяти

ENOENT

Некорректный режим движка рендеринга Data Engine

Примеры использования:

```

QApplication app(argc, argv);
gis_core_connection_t connection;
gis_core_request_parameters_t map;
gis_data_engine_context_t data_engine_ctx;
gis_render_rb_context_t rb_ctx;

gis_core_link_init( &connection );
gis_core_link_connect( &connection, 666 );
gis_core_databuffer_attach( &connection );
gis_core_request_parameters_init( &map );
gis_core_databuffer_data_request( &connection, &map, 1 );
gis_data_engine_alloc( 800, 600, GIS_DATA_ENGINE_MODE_BUFFER, GIS_DATA_ENGINE_BPP_32,
↪ &connection, &data_engine_ctx );

QPainter painter;
if ( gis_render_rb_alloc( &rb_ctx, data_engine_ctx, &painter ) != EOK )
{
    printf( "Failed to allocate RB context\n" );
    return 1;
}

gis_render_rb_free( &rb_ctx );
gis_data_engine_free( &data_engine_ctx );
gis_core_databuffer_detach();
gis_core_link_destroy( &connection, true );

```

Классификация:

ПК ЦКИ для ЗОСРВ «Нейтрино»

Безопасность использования	
Обработчик прерываний	Нет
Обработчик сигналов	Нет
В потоке	Да

Тематические ссылки:

[gis_render_rb_context_t](#), [gis_render_rb_draw\(\)](#), [gis_render_rb_free\(\)](#)

Поддерживается, начиная с ПК ЦКИ редакции 1.2

gis_render_rb_draw()

Рисование карты.

Прототип:

```
#include <gis/gis_renderbuffer.h>

int gis_render_rb_draw( gis_render_rb_context_t rb_ctx );
```

Аргументы:

rb_ctx

Контекст движка рендеринга Render Buffer.

Библиотека:

renderbuffer

Описание:

Функция вызывает отрисовку карты по установленным параметрам при помощи переданного QPainter.

Возвращаемое значение:

EOK

Успешное завершение

EFAULT

Рисование завершилось с ошибками

ETIME

Произошёл таймаут рисования

Примеры использования:

```
QApplication app(argc, argv);
gis_core_connection_t connection;
gis_core_request_parameters_t map;
gis_data_engine_context_t data_engine_ctx;
gis_render_rb_context_t rb_ctx;

gis_core_link_init( &connection );
gis_core_link_connect( &connection, 666 );
gis_core_databuffer_attach( &connection );
```

```

gis_core_request_parameters_init( &map );
gis_core_databuffer_data_request( &connection, &map, 1 );
gis_data_engine_alloc( 800, 600, GIS_DATA_ENGINE_MODE_BUFFER, GIS_DATA_ENGINE_BPP_32,
↪ &connection, &data_engine_ctx );

QPainter painter;
gis_render_rb_alloc( &rb_ctx, data_engine_ctx, &painter );

if ( gis_render_rb_draw(rb_ctx) != EOK )
{
    printf( "Failed in RB draw \n" );
    return 1;
}

gis_render_rb_free( &rb_ctx );
gis_data_engine_free( &data_engine_ctx );
gis_core_databuffer_detach();
gis_core_link_destroy( &connection, true );

```

Классификация:

ПК ЦКИ для ЗОСРВ «Нейтрино»

Безопасность использования	
Обработчик прерываний	Нет
Обработчик сигналов	Нет
В потоке	Да

Тематические ссылки:

`gis_render_rb_context_t`, `gis_render_rb_draw()`, `gis_render_rb_alloc()`

Поддерживается, начиная с ПК ЦКИ редакции 1.2

gis_render_rb_free()

Освобождение контекста движка рендеринга.

Прототип:

```
#include <gis/gis_renderbuffer.h>

int gis_render_rb_free( gis_render_rb_context_t *rb_ctx );
```

Аргументы:

rb_ctx

Указатель на контекст движка рендеринга Render Buffer.

Библиотека:

renderbuffer

Описание:

Функция освобождает контекст движка рендеринга Render Buffer.

Возвращаемое значение:

EOK

Успешное завершение

EINVAL

Некорректные параметры функции

Примеры использования:

```
QApplication app(argc, argv);
gis_core_connection_t connection;
gis_core_request_parameters_t map;
gis_data_engine_context_t data_engine_ctx;
gis_render_rb_context_t rb_ctx;

gis_core_link_init( &connection );
gis_core_link_connect( &connection, 666 );
gis_core_databuffer_attach( &connection );
gis_core_request_parameters_init( &map );
gis_core_databuffer_data_request( &connection, &map, 1 );
```

```

gis_data_engine_alloc( 800, 600, GIS_DATA_ENGINE_MODE_BUFFER, GIS_DATA_ENGINE_BPP_32,
↳ &connection, &data_engine_ctx );

QPainter painter;
gis_render_rb_alloc( &rb_ctx, data_engine_ctx, &painter );

if ( gis_render_rb_free( &rb_ctx ) != EOK )
{
    printf( "Failed to free RB context\n" );
    return 1;
}
gis_data_engine_free( &data_engine_ctx );
gis_core_databuffer_detach();
gis_core_link_destroy( &connection, true );

```

Классификация:

ПК ЦКИ для ЗОСРВ «Нейтрино»

Безопасность использования	
Обработчик прерываний	Нет
Обработчик сигналов	Нет
В потоке	Да

Тематические ссылки:

gis_render_rb_context_t, gis_render_rb_draw(), gis_render_rb_alloc()

Поддерживается, начиная с ПК ЦКИ редакции 1.2

gis_render_rb_context_t

Контекст движка рендеринга Render Buffer.

Формат:

```
#include <gis/gis_renderbuffer.h>

typedef void * gis_render_rb_context_t;
```

Описание:

Данный контекст обеспечивает управление параметрами движка рендеринга Render Buffer.

Классификация:

ПК ЦКИ для ЗОСРВ «Нейтрино»

Поддерживается, начиная с ПК ЦКИ редакции 1.2

gis_data_raw_ctx_alloc()

Создание контекста RAW для работы с областью данных.

Прототип:

```
#include <gis/gis_raw.h>

int gis_data_raw_ctx_alloc( gis_data_raw_context_t *ctx );
```

Аргументы:

ctx

Указатель на контекст `gis_data_raw_context_t`.

Библиотека:

gishelper

Описание:

Функция создаёт контекст RAW для работы с областью данных. Контекст должен быть создан до вызова прочих функций `gis_data_raw_*`.

Возвращаемое значение:

EOK

Успешное завершение.

EINVAL

Некорректный указатель на контекст.

ENOMEM

Недостаточно памяти для создания контекста.

Примеры использования:

```

gis_core_connection_t      connection;
gis_core_request_parameters_t  map;
gis_data_raw_context_t     raw_ctx;

gis_core_link_init( &connection );
gis_core_link_connect( &connection, 777 );
gis_core_databuffer_attach( &connection );
gis_core_request_parameters_init( &map );
gis_core_databuffer_data_request( &connection, &map, 1 );

gis_data_raw_ctx_alloc( &raw_ctx );

if ( gis_data_raw_ctx_import_databuffer( raw_ctx ) != EOK )
{
    printf( "Unable to import databuffer - probably bad alloc.\n" );
    gis_data_raw_ctx_free( &raw_ctx );
    gis_core_databuffer_detach();
    gis_core_link_destroy( &connection, true );
    return 1;
}

gis_data_raw_ctx_free( &raw_ctx );
gis_core_databuffer_detach();
gis_core_link_destroy( &connection, true );

```

Классификация:

ПК ЦКИ для ЗОСРВ «Нейтрино»

Безопасность использования	
Обработчик прерываний	Нет
Обработчик сигналов	Нет
В потоке	Нет

Тематические ссылки:

[gis_data_raw_context_t](#), [gis_data_raw_ctx_free\(\)](#)

gis_data_raw_ctx_free()

Удаление контекста RAW, необходимого для работы с областью данных.

Прототип:

```
#include <gis/gis_raw.h>

void gis_data_raw_ctx_free( gis_data_raw_context_t *ctx );
```

Аргументы:

ctx

Указатель на контекст `gis_data_raw_context_t`.

Библиотека:

gishelper

Описание:

Функция удаляет контекст RAW для работы с областью данных и обнуляет указатель.

Примеры использования:

```
gis_data_raw_context_t raw_ctx;
gis_data_raw_ctx_alloc( &raw_ctx );
gis_data_raw_ctx_free( &raw_ctx );
```

Классификация:

ПК ЦКИ для ЗОСРВ «Нейтрино»

Безопасность использования	
Обработчик прерываний	Нет
Обработчик сигналов	Нет
В потоке	Нет

Тематические ссылки:

`gis_data_raw_context_t`, `gis_data_raw_ctx_alloc()`

gis_data_raw_ctx_import_databuffer()

Получение указателя на разделяемую область памяти.

Прототип:

```
#include <gis/gis_raw.h>

int gis_data_raw_ctx_import_databuffer( gis_data_raw_context_t ctx );
```

Аргументы:

ctx

Контекст `gis_data_raw_context_t`.

Библиотека:

gishelper

Описание:

Функция получает указатель на разделяемую область памяти.

Возвращаемое значение:

EOK

Успешное завершение.

EINVAL

Некорректный контекст.

ENOMEM

Некорректный указатель на разделяемую область памяти.

Примеры использования:

```
gis_core_connection_t      connection;
gis_core_request_parameters_t map;
gis_data_raw_context_t     raw_ctx;

gis_core_link_init( &connection );
gis_core_link_connect( &connection, 777 );
gis_core_databuffer_attach( &connection );
gis_core_request_parameters_init( &map );
gis_core_databuffer_data_request( &connection, &map, 1 );
```

```

gis_data_raw_ctx_alloc( &raw_ctx );
gis_data_raw_ctx_import_databuffer( raw_ctx );

if ( gis_data_raw_select_first_map( raw_ctx ) != EOK )
{
    printf( "Import databuffer failed - can't select first map.\n" );
    gis_data_raw_ctx_free( &raw_ctx );
    gis_core_databuffer_detach();
    gis_core_link_destroy( &connection, true );
    return 1;
}

gis_data_raw_ctx_free( &raw_ctx );
gis_core_databuffer_detach();
gis_core_link_destroy( &connection, true );

```

Классификация:

ПК ЦКИ для ЗОСРВ «Нейтрино»

Безопасность использования	
Обработчик прерываний	Нет
Обработчик сигналов	Нет
В потоке	Нет

Тематические ссылки:

[gis_data_raw_context_t](#), [gis_data_raw_get_map_count\(\)](#)

gis_data_raw_get_map_count()

Получение количества карт в разделяемой области памяти.

Прототип:

```
#include <gis/gis_raw.h>

int gis_data_raw_get_map_count( gis_data_raw_context_t ctx );
```

Аргументы:

ctx

Контекст `gis_data_raw_context_t`.

Библиотека:

gishelper

Описание:

Функция возвращает количество карт в разделяемой области памяти.

Возвращаемое значение:

int

Количество карт.

(-EINVAL)

Некорректный указатель на контекст.

(-EBADFD)

Некорректный указатель на разделяемую область памяти.

Примеры использования:

```
gis_core_connection_t      connection;
gis_core_request_parameters_t map;
gis_data_raw_context_t     raw_ctx;

gis_core_link_init( &connection );
gis_core_link_connect( &connection, 777 );
gis_core_databuffer_attach( &connection );
gis_core_request_parameters_init( &map );
gis_core_databuffer_data_request( &connection, &map, 1 );
```

```

gis_data_raw_ctx_alloc( &raw_ctx );
gis_data_raw_ctx_import_databuffer( raw_ctx );

int count = gis_data_raw_get_map_count( raw_ctx );

if ( count != 3 )
{
    printf( "Incorrect number of maps in databuffer.\n" );
    gis_data_raw_ctx_free( &raw_ctx );
    gis_core_databuffer_detach();
    gis_core_link_destroy( &connection, true );
    return 1;
}

gis_data_raw_ctx_free( &raw_ctx );
gis_core_databuffer_detach();
gis_core_link_destroy( &connection, true );

```

Классификация:

ПК ЦКИ для ЗОСРВ «Нейтрино»

Безопасность использования	
Обработчик прерываний	Нет
Обработчик сигналов	Нет
В потоке	Нет

Тематические ссылки:

[gis_data_raw_context_t](#), [gis_data_raw_map_get_object_count\(\)](#)

gis_data_raw_get_raster_block_count()

Получение количества блоков растровых данных в текущей карте разделяемой области памяти

Прототип:

```
#include <gis/gis_raw.h>

int gis_data_raw_get_raster_block_count( const gis_data_raw_context_t raw_ctx, size_t
↪ *block_number );
```

Аргументы:

ctx

Контекст `gis_data_raw_context_t`.

block_number

Указатель на переменную, в которую запишется количество блоков растровой карты.

Библиотека:

gishelper

Описание:

Функция возвращает количество блоков в текущей выбранной растровой карте в разделяемой области памяти.

Возвращаемое значение:

EOK

Успешное завершение.

EINVAL

Некорректный аргумент.

EBADFD

Некорректный указатель на разделяемую область памяти.

ENODATA

Отсутствуют карты в разделяемой области памяти.

ENOMEM

Не удалось выделить память под растровый контекст.

EFAULT

Ошибка работы с растровой картой.

Примеры использования:

```
gis_core_connection_t      connection;
gis_core_request_parameters_t map;
gis_data_raw_context_t     raw_ctx;

gis_core_link_init( &connection );
gis_core_link_connect( &connection, 777 );
gis_core_databuffer_attach( &connection );
gis_core_request_parameters_init( &map );
gis_core_databuffer_data_request( &connection, &map, 1 );

gis_data_raw_ctx_alloc( &raw_ctx );
gis_data_raw_ctx_import_databuffer( raw_ctx );
gis_data_raw_select_first_map( raw_ctx );
do {
    if ( GIS_IS_CORE_MAP_DATA_SOURCE_VECTOR( gis_data_raw_map_get_data_source( raw_ctx )
↪ ) )
        continue;

    size_t block_number = 0;
    if ( gis_data_raw_get_raster_block_count( raw_ctx, &block_number ) != EOK )
    {
        gis_data_raw_ctx_free( &raw_ctx );
        gis_core_databuffer_detach();
        gis_core_link_destroy( &connection, true );
        return 1;
    }

    if ( block_number != 15 )
    {
        printf( "Incorrect number of blocks.\n" );
        gis_data_raw_ctx_free( &raw_ctx );
        gis_core_databuffer_detach();
        gis_core_link_destroy( &connection, true );
        return 1;
    }

} while ( gis_data_raw_select_next_map( raw_ctx ) == EOK);

gis_data_raw_ctx_free( &raw_ctx );
gis_core_databuffer_detach();
gis_core_link_destroy( &connection, true );
```

Классификация:

ПК ЦКИ для ЗОСРВ «Нейтрино»

Безопасность использования	
<i>Обработчик прерываний</i>	Нет
<i>Обработчик сигналов</i>	Нет
<i>В потоке</i>	Нет

Тематические ссылки:

`gis_data_raw_context_t`, `gis_data_raw_get_raster_width_in_blocks()`, `gis_data_raw_get_raster_height_in_blocks()`, `gis_data_raw_get_raster_width_in_elements()`, `gis_data_raw_get_raster_height_in_elements()`.

Поддерживается, начиная с ПК ЦКИ редакции 1.1.0

gis_data_raw_get_raster_block_height()

Получение высоты блока растровых данных

Прототип:

```
#include <gis/gis_raw.h>

int gis_data_raw_get_raster_block_height( const gis_data_raw_context_t raw_ctx, uint32_t
↪ index, uint32_t *block_height );
```

Аргументы:

ctx

Контекст `gis_data_raw_context_t`.

index

Последовательный индекс блока.

block_height

Указатель на переменную, в которую запишется высота указанного блока растровой карты в элементах.

Библиотека:

gishelper

Описание:

Функция возвращает высоту блока данных в текущей выбранной растровой карте в разделяемой области памяти.

Возвращаемое значение:

EOK

Успешное завершение.

EINVAL

Некорректный аргумент.

EBADFD

Некорректный указатель на разделяемую область памяти.

ENODATA

Отсутствуют карты в разделяемой области памяти.

ENOMEM

Не удалось выделить память под растровый контекст.

EFAULT

Ошибка работы с растровой картой.

Примеры использования:

```
gis_core_connection_t      connection;
gis_core_request_parameters_t map;
gis_data_raw_context_t     raw_ctx;

gis_core_link_init( &connection );
gis_core_link_connect( &connection, 777 );
gis_core_databuffer_attach( &connection );
gis_core_request_parameters_init( &map );
gis_core_databuffer_data_request( &connection, &map, 1 );

gis_data_raw_ctx_alloc( &raw_ctx );
gis_data_raw_ctx_import_databuffer( raw_ctx );
gis_data_raw_select_first_map( raw_ctx );

do {
    if ( GIS_IS_CORE_MAP_DATA_SOURCE_VECTOR( gis_data_raw_map_get_data_source( raw_ctx )
↪ ) )
        continue;

    size_t    block_number = 0;
    uint32_t  block_height = 0;
    gis_data_raw_get_raster_block_count( raw_ctx, &block_number );

    if ( gis_data_raw_get_raster_block_height( raw_ctx, 0, &block_height ) != EOK )
    {
        gis_data_raw_ctx_free( &raw_ctx );
        gis_core_databuffer_detach();
        gis_core_link_destroy( &connection, true );
        printf( "Error\n" );
        return 1;
    }

    if ( block_height != 256 )
    {
        gis_data_raw_ctx_free( &raw_ctx );
        gis_core_databuffer_detach();
        gis_core_link_destroy( &connection, true );
        printf( "Incorrect height: %d\n", block_height );
        return 1;
    }

} while ( gis_data_raw_select_next_map( raw_ctx ) == EOK);

gis_data_raw_ctx_free( &raw_ctx );
```

```
gis_core_databuffer_detach();
gis_core_link_destroy( &connection, true );
```

Классификация:

ПК ЦКИ для ЗОСРВ «Нейтрино»

Безопасность использования	
Обработчик прерываний	Нет
Обработчик сигналов	Нет
В потоке	Нет

Тематические ссылки:

gis_data_raw_context_t, gis_data_raw_get_raster_block_count(), gis_data_raw_get_raster_block_width(), gis_data_raw_get_raster_width_in_blocks(), gis_data_raw_get_raster_height_in_blocks(), gis_data_raw_get_raster_width_in_elements(), gis_data_raw_get_raster_height_in_elements().

Поддерживается, начиная с ПК ЦКИ редакции 1.1.0

gis_data_raw_get_raster_block_upper_left_corner()

Получение координат верхнего левого угла указанного блока растровой карты (метры в проекции растра)

Прототип:

```
#include <gis/gis_raw.h>

int gis_data_raw_get_raster_block_upper_left_corner( const gis_data_raw_context_t raw_ctx,
↳ uint32_t index, double_point_t *pULCorner );
```

Аргументы:

ctx

Контекст gis_data_raw_context_t.

ctx

Индекс блока.

pULCorner

Указатель на переменную, в которую запишутся координаты верхнего левого угла растра в метрах.

Библиотека:

gishelper

Описание:

Функция возвращает координаты верхнего левого угла указанного блока растровой карты (метры в исходной проекции растра)

Возвращаемое значение:

EOK

Успешное завершение.

EINVAL

Некорректный аргумент.

EBADFD

Некорректный указатель на разделяемую область памяти.

ENODATA

Отсутствуют карты в разделяемой области памяти.

ENOMEM

Не удалось выделить память под растровый контекст.

EFAULT

Ошибка работы с растровой картой.

Примеры использования:

```
gis_core_connection_t      connection;
gis_core_request_parameters_t map;
gis_data_raw_context_t     raw_ctx;

gis_core_link_init( &connection );
gis_core_link_connect( &connection, 777 );
gis_core_databuffer_attach( &connection );
gis_core_request_parameters_init( &map );
gis_core_databuffer_data_request( &connection, &map, 1 );

gis_data_raw_ctx_alloc( &raw_ctx );
gis_data_raw_ctx_import_databuffer( raw_ctx );
gis_data_raw_select_first_map( raw_ctx );

do {
    if ( GIS_IS_CORE_MAP_DATA_SOURCE_VECTOR( gis_data_raw_map_get_data_source( raw_ctx )
↪ ) )
        continue;

    size_t block_number = 0;
    gis_data_raw_get_raster_block_count( raw_ctx, &block_number );

    double_point_t ul;
    if ( gis_data_raw_get_raster_block_upper_left_corner( raw_ctx, 0, &ul ) != EOK )
    {
        gis_data_raw_ctx_free( &raw_ctx );
        gis_core_databuffer_detach();
        gis_core_link_destroy( &connection, true );
        printf( "Error\n" );
        return 1;
    }
} while ( gis_data_raw_select_next_map( raw_ctx ) == EOK);

gis_data_raw_ctx_free( &raw_ctx );
gis_core_databuffer_detach();
gis_core_link_destroy( &connection, true );
```

Классификация:

ПК ЦКИ для ЗОСРВ «Нейтрино»

Безопасность использования	
<i>Обработчик прерываний</i>	Нет
<i>Обработчик сигналов</i>	Нет
<i>В потоке</i>	Нет

Тематические ссылки:

`gis_data_raw_context_t`, `gis_data_raw_get_raster_meters_per_element()`, `gis_data_raw_get_raster_upper_left_corner()`

Поддерживается, начиная с ПК ЦКИ редакции 1.1.0

gis_data_raw_get_raster_block_width()

Получение ширины блока растровых данных

Прототип:

```
#include <gis/gis_raw.h>

int gis_data_raw_get_raster_block_width( const gis_data_raw_context_t raw_ctx, uint32_t
↪ index, uint32_t *block_width );
```

Аргументы:

ctx

Контекст `gis_data_raw_context_t`.

index

Последовательный индекс блока.

block_width

Указатель на переменную, в которую запишется ширина указанного блока растровой карты в элементах.

Библиотека:

gishelper

Описание:

Функция возвращает ширину блока данных в текущей выбранной растровой карте в разделяемой области памяти.

Возвращаемое значение:

EOK

Успешное завершение.

EINVAL

Некорректный аргумент.

EBADFD

Некорректный указатель на разделяемую область памяти.

ENODATA

Отсутствуют карты в разделяемой области памяти.

ENOMEM

Не удалось выделить память под растровый контекст.

EFAULT

Ошибка работы с растровой картой.

Примеры использования:

```
gis_core_connection_t      connection;
gis_core_request_parameters_t  map;
gis_data_raw_context_t      raw_ctx;

gis_core_link_init( &connection );
gis_core_link_connect( &connection, 777 );
gis_core_databuffer_attach( &connection );
gis_core_request_parameters_init( &map );
gis_core_databuffer_data_request( &connection, &map, 1 );

gis_data_raw_ctx_alloc( &raw_ctx );
gis_data_raw_ctx_import_databuffer( raw_ctx );
gis_data_raw_select_first_map( raw_ctx );

do {
    if ( GIS_IS_CORE_MAP_DATA_SOURCE_VECTOR( gis_data_raw_map_get_data_source( raw_ctx )
↪ ) )
        continue;

    size_t    block_number = 0;
    uint32_t  block_width = 0;
    gis_data_raw_get_raster_block_count( raw_ctx, &block_number );

    if ( gis_data_raw_get_raster_block_width( raw_ctx, 0, &block_width ) != EOK )
    {
        gis_data_raw_ctx_free( &raw_ctx );
        gis_core_databuffer_detach();
        gis_core_link_destroy( &connection, true );
        printf( "Error\n" );
        return 1;
    }

    if ( block_width != 256 )
    {
        gis_data_raw_ctx_free( &raw_ctx );
        gis_core_databuffer_detach();
        gis_core_link_destroy( &connection, true );
        printf( "Incorrect width: %d\n", block_width );
        return 1;
    }

} while ( gis_data_raw_select_next_map( raw_ctx ) == EOK);

gis_data_raw_ctx_free( &raw_ctx );
```

```
gis_core_databuffer_detach();
gis_core_link_destroy( &connection, true );
```

Классификация:

ПК ЦКИ для ЗОСРВ «Нейтрино»

Безопасность использования	
Обработчик прерываний	Нет
Обработчик сигналов	Нет
В потоке	Нет

Тематические ссылки:

gis_data_raw_context_t, gis_data_raw_get_raster_block_count(), gis_data_raw_get_raster_block_height(), gis_data_raw_get_raster_width_in_blocks(), gis_data_raw_get_raster_height_in_blocks(), gis_data_raw_get_raster_width_in_elements(), gis_data_raw_get_raster_height_in_elements().

Поддерживается, начиная с ПК ЦКИ редакции 1.1.0

gis_data_raw_get_raster_data()

Получение данных блока растровых данных

Прототип:

```
#include <gis/gis_raw.h>

int gis_data_raw_get_raster_data( const gis_data_raw_context_t ctx, uint32_t index, double
↵ *block_data );
```

Аргументы:

ctx
Контекст gis_data_raw_context_t.

index
Последовательный индекс блока.

block_data
Указатель на буффер.

Библиотека:

gishelper

Описание:

Функция заполняет массив данных block_data данными из указанного блока текущей выбранной растровой карты в разделяемой области памяти.

Возвращаемое значение:

EOK
Успешное завершение.

EINVAL
Некорректный аргумент.

EBADFD
Некорректный указатель на разделяемую область памяти.

ENODATA
Отсутствуют карты в разделяемой области памяти.

ENOMEM
Не удалось выделить память под растровый контекст.

EFAULT

Ошибка работы с растровой картой.

Примеры использования:

```
gis_core_connection_t      connection;
gis_core_request_parameters_t map;
gis_data_raw_context_t     raw_ctx;

gis_core_link_init( &connection );
gis_core_link_connect( &connection, 777 );
gis_core_databuffer_attach( &connection );
gis_core_request_parameters_init( &map );
gis_core_databuffer_data_request( &connection, &map, 1 );

gis_data_raw_ctx_alloc( &raw_ctx );
gis_data_raw_ctx_import_databuffer( raw_ctx );
gis_data_raw_select_first_map( raw_ctx );

do {
    if ( ! GIS_IS_CORE_MAP_DATA_SOURCE_VECTOR( gis_data_raw_map_get_data_source( raw_ctx
↵ ) ) )
        break;
} while ( gis_data_raw_select_next_map( raw_ctx ) == EOK);

size_t    block_number = 0;

gis_data_raw_get_raster_block_count( raw_ctx, &block_number );

uint32_t block_width = 0;
uint32_t block_height = 0;
gis_data_raw_get_raster_block_height( raw_ctx, 0, &block_height );
gis_data_raw_get_raster_block_width( raw_ctx, 0, &block_width );

size_t block_size = block_width * block_height;
double *block_data = (double*)calloc( block_size, sizeof( *block_data ) );
if ( !block_data )
{
    printf( "Failed to allocate memory\n" );
    gis_data_raw_ctx_free( &raw_ctx );
    gis_core_databuffer_detach();
    gis_core_link_destroy( &connection, true );
    return 1;
}

if ( gis_data_raw_get_raster_data( raw_ctx, 0, block_data ) != EOK )
{
    printf( "Failed to get data\n" );
    free( block_data );
}
```

```

        gis_data_raw_ctx_free( &raw_ctx );
        gis_core_databuffer_detach();
        gis_core_link_destroy( &connection, true );
    return 1;
}

printf( "%lf %lf %lf %lf", *block_data, *(block_data+1), *(block_data+2),
↪  *(block_data+3) );

free( block_data );
gis_data_raw_ctx_free( &raw_ctx );
gis_core_databuffer_detach();
gis_core_link_destroy( &connection, true );

```

Классификация:

ПК ЦКИ для ЗОСРВ «Нейтрино»

Безопасность использования	
Обработчик прерываний	Нет
Обработчик сигналов	Нет
В потоке	Нет

Тематические ссылки:

gis_data_raw_context_t, gis_data_raw_get_raster_block_count(), gis_data_raw_get_raster_block_height().

Поддерживается, начиная с ПК ЦКИ редакции 1.1.0

gis_data_raw_get_raster_height_in_blocks()

Получение высоты текущей растровой карты в блоках

Прототип:

```
#include <gis/gis_raw.h>

int gis_data_raw_get_raster_height_in_blocks( const gis_data_raw_context_t raw_ctx,
↪ uint32_t *pHeight );
```

Аргументы:

ctx

Контекст `gis_data_raw_context_t`.

pHeight

Указатель на переменную, в которую запишется количество блоков в карте в высоту.

Библиотека:

gishelper

Описание:

Функция возвращает высоту текущей растровой карты в блоках.

Возвращаемое значение:

EOK

Успешное завершение.

EINVAL

Некорректный аргумент.

EBADFD

Некорректный указатель на разделяемую область памяти.

ENODATA

Отсутствуют карты в разделяемой области памяти.

ENOMEM

Не удалось выделить память под растровый контекст.

EFAULT

Ошибка работы с растровой картой.

Примеры использования:

```
gis_core_connection_t      connection;
gis_core_request_parameters_t map;
gis_data_raw_context_t     raw_ctx;

gis_core_link_init( &connection );
gis_core_link_connect( &connection, 777 );
gis_core_databuffer_attach( &connection );
gis_core_request_parameters_init( &map );
gis_core_databuffer_data_request( &connection, &map, 1 );

gis_data_raw_ctx_alloc( &raw_ctx );
gis_data_raw_ctx_import_databuffer( raw_ctx );
gis_data_raw_select_first_map( raw_ctx );
do {
    if ( GIS_IS_CORE_MAP_DATA_SOURCE_VECTOR( gis_data_raw_map_get_data_source( raw_ctx )
↪ ) )
        continue;

    uint32_t height = 0;
    if ( gis_data_raw_get_raster_height_in_blocks( raw_ctx, &height ) != EOK ) {
        gis_data_raw_ctx_free( &raw_ctx );
        gis_core_databuffer_detach();
        gis_core_link_destroy( &connection, true );
        return 1;
    }

    if ( height != 3 )
    {
        printf( "Incorrect width.\n" );
        gis_data_raw_ctx_free( &raw_ctx );
        gis_core_databuffer_detach();
        gis_core_link_destroy( &connection, true );
        return 1;
    }

} while ( gis_data_raw_select_next_map( raw_ctx ) == EOK);

gis_data_raw_ctx_free( &raw_ctx );
gis_core_databuffer_detach();
gis_core_link_destroy( &connection, true );
```

Классификация:

ПК ЦКИ для ЗОСРВ «Нейтрино»

Безопасность использования	
<i>Обработчик прерываний</i>	Нет
<i>Обработчик сигналов</i>	Нет
<i>В потоке</i>	Нет

Тематические ссылки:

`gis_data_raw_context_t`, `gis_data_raw_get_raster_width_in_blocks()`, `gis_data_raw_get_raster_width_in_elements()`, `gis_data_raw_get_raster_height_in_elements()`.

Поддерживается, начиная с ПК ЦКИ редакции 1.1.0

gis_data_raw_get_raster_height_in_elements()

Получение высоты растровой карты (в элементах)

Прототип:

```
#include <gis/gis_raw.h>

int gis_data_raw_get_raster_height_in_elements( const gis_data_raw_context_t ctx, uint32_t
↪ *pHeight );
```

Аргументы:

ctx

Контекст `gis_data_raw_context_t`.

pHeight

Указатель на переменную, в которую запишется высота растровой карты в элементах.

Библиотека:

gishelper

Описание:

Функция возвращает высоту текущей выбранной растровой карты в элементах.

Возвращаемое значение:

EOK

Успешное завершение.

EINVAL

Некорректный аргумент.

EBADFD

Некорректный указатель на разделяемую область памяти.

ENODATA

Отсутствуют карты в разделяемой области памяти.

ENOMEM

Не удалось выделить память под растровый контекст.

EFAULT

Ошибка работы с растровой картой.

Примеры использования:

```
gis_core_connection_t      connection;
gis_core_request_parameters_t map;
gis_data_raw_context_t     raw_ctx;

gis_core_link_init( &connection );
gis_core_link_connect( &connection, 777 );
gis_core_databuffer_attach( &connection );
gis_core_request_parameters_init( &map );
gis_core_databuffer_data_request( &connection, &map, 1 );

gis_data_raw_ctx_alloc( &raw_ctx );
gis_data_raw_ctx_import_databuffer( raw_ctx );
gis_data_raw_select_first_map( raw_ctx );

do {
    if ( GIS_IS_CORE_MAP_DATA_SOURCE_VECTOR( gis_data_raw_map_get_data_source( raw_ctx )
↵ ) )
        continue;

    uint32_t height = 0;
    size_t block_number = 0;
    gis_data_raw_get_raster_block_count( raw_ctx, &block_number );

    if ( gis_data_raw_get_raster_height_in_elements( raw_ctx, &height ) != EOK )
    {
        gis_data_raw_ctx_free( &raw_ctx );
        gis_core_databuffer_detach();
        gis_core_link_destroy( &connection, true );
        printf( "Error\n" );
        return 1;
    }

    if ( height != 685 )
    {
        gis_data_raw_ctx_free( &raw_ctx );
        gis_core_databuffer_detach();
        gis_core_link_destroy( &connection, true );
        printf( "Incorrect height: %d\n", height );
        return 1;
    }

} while ( gis_data_raw_select_next_map( raw_ctx ) == EOK);

gis_data_raw_ctx_free( &raw_ctx );
gis_core_databuffer_detach();
gis_core_link_destroy( &connection, true );
```

Классификация:

ПК ЦКИ для ЗОСРВ «Нейтрино»

Безопасность использования	
Обработчик прерываний	Нет
Обработчик сигналов	Нет
В потоке	Нет

Тематические ссылки:

`gis_data_raw_context_t`, `gis_data_raw_get_raster_block_count()`, `gis_data_raw_get_raster_block_height()`, `gis_data_raw_get_raster_height_in_blocks()`, `gis_data_raw_get_raster_width_in_elements()`.

Поддерживается, начиная с ПК ЦКИ редакции 1.1.0

gis_data_raw_get_raster_meters_per_element()

Получение разрешения растровой карты (метры на элемент)

Прототип:

```
#include <gis/gis_raw.h>

int gis_data_raw_get_raster_meters_per_element( const gis_data_raw_context_t ctx, double
↵ *pMetersPerElement );
```

Аргументы:

ctx

Контекст `gis_data_raw_context_t`.

pMetersPerElement

Указатель на переменную, в которую запишется разрешение растровой карты.

Библиотека:

gishelper

Описание:

Функция возвращает разрешение растровой карты (количество метров на элемент).

Возвращаемое значение:

EOK

Успешное завершение.

EINVAL

Некорректный аргумент.

EBADFD

Некорректный указатель на разделяемую область памяти.

ENODATA

Отсутствуют карты в разделяемой области памяти.

ENOMEM

Не удалось выделить память под растровый контекст.

EFAULT

Ошибка работы с растровой картой.

Примеры использования:

```
gis_core_connection_t      connection;
gis_core_request_parameters_t map;
gis_data_raw_context_t     raw_ctx;

gis_core_link_init( &connection );
gis_core_link_connect( &connection, 777 );
gis_core_databuffer_attach( &connection );
gis_core_request_parameters_init( &map );
gis_core_databuffer_data_request( &connection, &map, 1 );

gis_data_raw_ctx_alloc( &raw_ctx );
gis_data_raw_ctx_import_databuffer( raw_ctx );
gis_data_raw_select_first_map( raw_ctx );

do {
    if ( GIS_IS_CORE_MAP_DATA_SOURCE_VECTOR( gis_data_raw_map_get_data_source( raw_ctx )
↪ ) )
        continue;

    double m_per_element = 0;
    size_t block_number = 0;
    gis_data_raw_get_raster_block_count( raw_ctx, &block_number );

    if ( gis_data_raw_get_raster_meters_per_element( raw_ctx, &m_per_element ) != EOK )
    {
        gis_data_raw_ctx_free( &raw_ctx );
        gis_core_databuffer_detach();
        gis_core_link_destroy( &connection, true );
        printf( "Error\n" );
        return 1;
    }

    if ( m_per_element != 250.0 )
    {
        gis_data_raw_ctx_free( &raw_ctx );
        gis_core_databuffer_detach();
        gis_core_link_destroy( &connection, true );
        printf( "Incorrect m/elem: %f\n", m_per_element );
        return 1;
    }

} while ( gis_data_raw_select_next_map( raw_ctx ) == EOK);

gis_data_raw_ctx_free( &raw_ctx );
gis_core_databuffer_detach();
gis_core_link_destroy( &connection, true );
```

Классификация:

ПК ЦКИ для ЗОСРВ «Нейтрино»

Безопасность использования	
<i>Обработчик прерываний</i>	Нет
<i>Обработчик сигналов</i>	Нет
<i>В потоке</i>	Нет

Тематические ссылки:

`gis_data_raw_context_t`, `gis_data_raw_get_raster_upper_left_corner()`, `gis_data_raw_get_raster_block_upper_left_corner()`

Поддерживается, начиная с ПК ЦКИ редакции 1.1.0

gis_data_raw_get_raster_upper_left_corner()

Получение координат верхнего левого угла растровой карты (метры в проекции растра)

Прототип:

```
#include <gis/gis_raw.h>

int gis_data_raw_get_raster_upper_left_corner( const gis_data_raw_context_t ctx,
↪ double_point_t *pULCorner );
```

Аргументы:

ctx

Контекст `gis_data_raw_context_t`.

pULCorner

Указатель на переменную, в которую запишутся координаты верхнего левого угла растра в метрах.

Библиотека:

gishelper

Описание:

Функция возвращает координаты верхнего левого угла растровой карты (метры в исходной проекции растра)

Возвращаемое значение:

EOK

Успешное завершение.

EINVAL

Некорректный аргумент.

EBADFD

Некорректный указатель на разделяемую область памяти.

ENODATA

Отсутствуют карты в разделяемой области памяти.

ENOMEM

Не удалось выделить память под растровый контекст.

EFAULT

Ошибка работы с растровой картой.

Примеры использования:

```
gis_core_connection_t      connection;
gis_core_request_parameters_t map;
gis_data_raw_context_t     raw_ctx;

gis_core_link_init( &connection );
gis_core_link_connect( &connection, 777 );
gis_core_databuffer_attach( &connection );
gis_core_request_parameters_init( &map );
gis_core_databuffer_data_request( &connection, &map, 1 );

gis_data_raw_ctx_alloc( &raw_ctx );
gis_data_raw_ctx_import_databuffer( raw_ctx );
gis_data_raw_select_first_map( raw_ctx );

do {
    if ( GIS_IS_CORE_MAP_DATA_SOURCE_VECTOR( gis_data_raw_map_get_data_source( raw_ctx )
↵ ) )
        continue;

    size_t block_number = 0;
    gis_data_raw_get_raster_block_count( raw_ctx, &block_number );

    double_point_t ul;
    if ( gis_data_raw_get_raster_upper_left_corner( raw_ctx, &ul ) != EOK )
    {
        gis_data_raw_ctx_free( &raw_ctx );
        gis_core_databuffer_detach();
        gis_core_link_destroy( &connection, true );
        printf( "Error\n" );
        return 1;
    }
} while ( gis_data_raw_select_next_map( raw_ctx ) == EOK);

gis_data_raw_ctx_free( &raw_ctx );
gis_core_databuffer_detach();
gis_core_link_destroy( &connection, true );
```

Классификация:

ПК ЦКИ для ЗОСРВ «Нейтрино»

Безопасность использования	
Обработчик прерываний	Нет
Обработчик сигналов	Нет
В потоке	Нет

Тематические ссылки:

`gis_data_raw_context_t`, `gis_data_raw_get_raster_meters_per_element()`, `gis_data_raw_get_raster_block_upper_left_c`

Поддерживается, начиная с ПК ЦКИ редакции 1.1.0

gis_data_raw_get_raster_width_in_blocks()

Получение ширины текущей растровой карты в блоках

Прототип:

```
#include <gis/gis_raw.h>

int gis_data_raw_get_raster_width_in_blocks( const gis_data_raw_context_t ctx, uint32_t
↪ *pWidth );
```

Аргументы:

ctx

Контекст `gis_data_raw_context_t`.

pWidth

Указатель на переменную, в которую запишется количество блоков в карте в ширину.

Библиотека:

gishelper

Описание:

Функция возвращает ширину текущей растровой карты в блоках.

Возвращаемое значение:

EOK

Успешное завершение.

EINVAL

Некорректный аргумент.

EBADFD

Некорректный указатель на разделяемую область памяти.

ENODATA

Отсутствуют карты в разделяемой области памяти.

ENOMEM

Не удалось выделить память под растровый контекст.

EFAULT

Ошибка работы с растровой картой.

Примеры использования:

```
gis_core_connection_t      connection;
gis_core_request_parameters_t map;
gis_data_raw_context_t     raw_ctx;

gis_core_link_init( &connection );
gis_core_link_connect( &connection, 777 );
gis_core_databuffer_attach( &connection );
gis_core_request_parameters_init( &map );
gis_core_databuffer_data_request( &connection, &map, 1 );

gis_data_raw_ctx_alloc( &raw_ctx );
gis_data_raw_ctx_import_databuffer( raw_ctx );
gis_data_raw_select_first_map( raw_ctx );
do {
    if ( GIS_IS_CORE_MAP_DATA_SOURCE_VECTOR( gis_data_raw_map_get_data_source( raw_ctx )
↪ ) )
        continue;

    uint32_t width = 0;
    if ( gis_data_raw_get_raster_width_in_blocks( raw_ctx, &width ) != EOK ) {
        gis_data_raw_ctx_free( &raw_ctx );
        gis_core_databuffer_detach();
        gis_core_link_destroy( &connection, true );
        return 1;
    }

    if ( width != 5 )
    {
        printf( "Incorrect width.\n" );
        gis_data_raw_ctx_free( &raw_ctx );
        gis_core_databuffer_detach();
        gis_core_link_destroy( &connection, true );
        return 1;
    }

} while ( gis_data_raw_select_next_map( raw_ctx ) == EOK);

gis_data_raw_ctx_free( &raw_ctx );
gis_core_databuffer_detach();
gis_core_link_destroy( &connection, true );
```

Классификация:

ПК ЦКИ для ЗОСРВ «Нейтрино»

Безопасность использования	
<i>Обработчик прерываний</i>	Нет
<i>Обработчик сигналов</i>	Нет
<i>В потоке</i>	Нет

Тематические ссылки:

`gis_data_raw_context_t`, `gis_data_raw_get_raster_height_in_blocks()`, `gis_data_raw_get_raster_width_in_elements()`, `gis_data_raw_get_raster_height_in_elements()`.

Поддерживается, начиная с ПК ЦКИ редакции 1.1.0

gis_data_raw_get_raster_width_in_elements()

Получение ширины растровой карты (в элементах)

Прототип:

```
#include <gis/gis_raw.h>

int gis_data_raw_get_raster_width_in_elements( const gis_data_raw_context_t ctx, uint32_t
↪ *pWidth );
```

Аргументы:

ctx

Контекст `gis_data_raw_context_t`.

pWidth

Указатель на переменную, в которую запишется ширина растровой карты в элементах.

Библиотека:

gishelper

Описание:

Функция возвращает ширину текущей выбранной растровой карты в элементах.

Возвращаемое значение:

EOK

Успешное завершение.

EINVAL

Некорректный аргумент.

EBADFD

Некорректный указатель на разделяемую область памяти.

ENODATA

Отсутствуют карты в разделяемой области памяти.

ENOMEM

Не удалось выделить память под растровый контекст.

EFAULT

Ошибка работы с растровой картой.

Примеры использования:

```
gis_core_connection_t      connection;
gis_core_request_parameters_t map;
gis_data_raw_context_t     raw_ctx;

gis_core_link_init( &connection );
gis_core_link_connect( &connection, 777 );
gis_core_databuffer_attach( &connection );
gis_core_request_parameters_init( &map );
gis_core_databuffer_data_request( &connection, &map, 1 );

gis_data_raw_ctx_alloc( &raw_ctx );
gis_data_raw_ctx_import_databuffer( raw_ctx );
gis_data_raw_select_first_map( raw_ctx );

do {
    if ( GIS_IS_CORE_MAP_DATA_SOURCE_VECTOR( gis_data_raw_map_get_data_source( raw_ctx )
↵ ) )
        continue;

    size_t    block_number = 0;
    uint32_t  width = 0;
    gis_data_raw_get_raster_block_count( raw_ctx, &block_number );

    if ( gis_data_raw_get_raster_width_in_elements( raw_ctx, &width ) != EOK )
    {
        gis_data_raw_ctx_free( &raw_ctx );
        gis_core_databuffer_detach();
        gis_core_link_destroy( &connection, true );
        printf( "Error\n" );
        return 1;
    }

    if ( width != 1044 )
    {
        gis_data_raw_ctx_free( &raw_ctx );
        gis_core_databuffer_detach();
        gis_core_link_destroy( &connection, true );
        printf( "Incorrect width: %d\n", width );
        return 1;
    }

} while ( gis_data_raw_select_next_map( raw_ctx ) == EOK);

gis_data_raw_ctx_free( &raw_ctx );
gis_core_databuffer_detach();
gis_core_link_destroy( &connection, true );
```

Классификация:

ПК ЦКИ для ЗОСРВ «Нейтрино»

Безопасность использования	
<i>Обработчик прерываний</i>	Нет
<i>Обработчик сигналов</i>	Нет
<i>В потоке</i>	Нет

Тематические ссылки:

`gis_data_raw_context_t`, `gis_data_raw_get_raster_block_width()`, `gis_data_raw_get_raster_width_in_blocks()`, `gis_data_raw_get_raster_height_in_elements()`.

Поддерживается, начиная с ПК ЦКИ редакции 1.1.0

gis_data_raw_map_get_borders()

Получение градусных границ текущей карты в разделяемой области памяти

Прототип:

```
#include <gis/gis_raw.h>

int gis_data_raw_map_get_borders( const gis_data_raw_context_t ctx, gis_borders_t *pBorders
↵ );
```

Аргументы:

ctx
Контекст `gis_data_raw_context_t`.

pBorders
Указатель на поле проекции `gis_borders_t`.

Библиотека:

gishelper

Описание:

Функция возвращает градусные границы текущей выбранной карты в разделяемой области памяти.

Возвращаемое значение:

EOK
Успешное завершение.

EINVAL
Некорректный аргумент.

EBADFD
Некорректный указатель на разделяемую область памяти.

ENODATA
Отсутствуют карты в разделяемой области памяти.

EFAULT
Некорректный указатель на текущую карту в разделяемой области памяти.

Примеры использования:

```

gis_core_connection_t      connection;
gis_core_request_parameters_t  map;
gis_data_raw_context_t     raw_ctx;

gis_core_link_init( &connection );
gis_core_link_connect( &connection, 777 );
gis_core_databuffer_attach( &connection );
gis_core_request_parameters_init( &map );
gis_core_databuffer_data_request( &connection, &map, 1 );

gis_data_raw_ctx_alloc( &raw_ctx );
gis_data_raw_ctx_import_databuffer( raw_ctx );
gis_data_raw_select_first_map( raw_ctx );

gis_borders_t borders;
if ( gis_data_raw_map_get_borders( raw_ctx, &borders ) != EOK ) {

    gis_data_raw_ctx_free( &raw_ctx );
    gis_core_databuffer_detach();
    gis_core_link_destroy( &connection, true );
    return -1;
}

gis_data_raw_ctx_free( &raw_ctx );
gis_core_databuffer_detach();
gis_core_link_destroy( &connection, true );

```

Классификация:

ПК ЦКИ для ЗОСРВ «Нейтрино»

Безопасность использования	
Обработчик прерываний	Нет
Обработчик сигналов	Нет
В потоке	Нет

Тематические ссылки:

gis_data_raw_context_t, gis_data_raw_select_first_map().

Поддерживается, начиная с ПК ЦКИ редакции 1.1.0

gis_data_raw_map_get_data_source()

Получение типа выбранной карты.

Прототип:

```
#include <gis/gis_raw.h>

gis_core_map_data_source_t gis_data_raw_map_get_data_source( gis_data_raw_context_t ctx );
```

Аргументы:

ctx

Контекст `gis_data_raw_context_t`.

Библиотека:

gishelper

Описание:

Функция возвращает тип текущей выбранной карты разделяемой области памяти.

Возвращаемое значение:

int

Успешное завершение.

`GIS_CORE_MAP_DATA_SOURCE_NONE`

Некорректный указатель на разделяемую область памяти.

Примеры использования:

```
gis_core_connection_t      connection;
gis_core_request_parameters_t map;
gis_data_raw_context_t     raw_ctx;

gis_core_link_init( &connection );
gis_core_link_connect( &connection, 777 );
gis_core_databuffer_attach( &connection );
gis_core_request_parameters_init( &map );
gis_core_databuffer_data_request( &connection, &map, 1 );

gis_data_raw_ctx_alloc( &raw_ctx );
```

```

gis_data_raw_ctx_import_databuffer( raw_ctx );

    gis_core_map_list_t map_list;
    gis_core_map_list_init( &map_list );
    gis_core_request_maps_list( &connection, &map, GIS_CORE_DRIVERS_SXF_LOCAL, &map_list );

    gis_data_raw_select_map( raw_ctx, GIS_CORE_DRIVERS_SXF_LOCAL, map_list.entry_list[0].id
↵ );

    gis_core_map_data_source_t res = gis_data_raw_map_get_data_source( raw_ctx );

    if ( res != GIS_CORE_MAP_DATA_SOURCE_SXF ) {

        printf( "Map source for first map is incorrect: get %d, expected: %d.",
↵
↵ res, GIS_CORE_MAP_DATA_SOURCE_SXF );
        gis_data_raw_ctx_free( &raw_ctx );
        gis_core_map_list_free( &map_list );
        gis_core_databuffer_detach();
        gis_core_link_destroy( &connection, true );
        return 1;
    }

    gis_data_raw_ctx_free( &raw_ctx );
    gis_core_map_list_free( &map_list );
    gis_core_databuffer_detach();
    gis_core_link_destroy( &connection, true );

```

Классификация:

ПК ЦКИ для ЗОСРВ «Нейтрино»

Безопасность использования	
Обработчик прерываний	Нет
Обработчик сигналов	Нет
В потоке	Нет

Тематические ссылки:

[gis_data_raw_context_t](#), [gis_core_map_data_source_t](#)

gis_data_raw_map_get_filename()

Получение названия текущей карты в разделяемой области памяти

Прототип:

```
#include <gis/gis_raw.h>

int gis_data_raw_map_get_filename( const gis_data_raw_context_t ctx, char *pMapFilename );
```

Аргументы:

ctx

Контекст gis_data_raw_context_t.

pMapFilename

Указатель на строку, в которую запишется название файла карты.

Библиотека:

gishelper

Описание:

Функция возвращает название текущей выбранной карты в разделяемой области памяти.

Возвращаемое значение:

EOK

Успешное завершение.

EINVAL

Некорректный аргумент.

EBADFD

Некорректный указатель на разделяемую область памяти.

ENODATA

Отсутствуют карты в разделяемой области памяти.

EFAULT

Некорректный указатель на текущую карту в разделяемой области памяти.

Примеры использования:

```

gis_core_connection_t      connection;
gis_core_request_parameters_t  map;
gis_data_raw_context_t     raw_ctx;

gis_core_link_init( &connection );
gis_core_link_connect( &connection, 777 );
gis_core_databuffer_attach( &connection );
gis_core_request_parameters_init( &map );
gis_core_databuffer_data_request( &connection, &map, 1 );

gis_data_raw_ctx_alloc( &raw_ctx );
gis_data_raw_ctx_import_databuffer( raw_ctx );

gis_data_raw_select_first_map( raw_ctx );

char map_filename[GIS_MAX_NAME_LENGTH];
if ( gis_data_raw_map_get_filename( raw_ctx, map_filename ) != EOK ) {

    gis_data_raw_ctx_free( &raw_ctx );
    gis_core_databuffer_detach();
    gis_core_link_destroy( &connection, true );
    return -1;
}

gis_data_raw_ctx_free( &raw_ctx );
gis_core_databuffer_detach();
gis_core_link_destroy( &connection, true );

```

Классификация:

ПК ЦКИ для ЗОСРВ «Нейтрино»

Безопасность использования	
Обработчик прерываний	Нет
Обработчик сигналов	Нет
В потоке	Нет

Тематические ссылки:

[gis_data_raw_context_t](#), [gis_data_raw_select_first_map\(\)](#)

Поддерживается, начиная с ПК ЦКИ редакции 1.1.0

gis_data_raw_map_get_object_count()

Получение количества объектов в текущей выбранной карте разделяемой области памяти.

Прототип:

```
#include <gis/gis_raw.h>

int gis_data_raw_map_get_object_count( gis_data_raw_context_t ctx );
```

Аргументы:

ctx

Контекст `gis_data_raw_context_t`.

Библиотека:

gishelper

Описание:

Функция возвращает количество объектов в текущей выбранной карте разделяемой области памяти.

Возвращаемое значение:

int

Количество объектов.

(-EFAULT)

Некорректный указатель на карту в разделяемой области памяти.

(-ENODATA)

Отсутствуют карты в разделяемой области памяти.

(-EBADFD)

Некорректный указатель на разделяемую область памяти.

(-EINVAL)

Некорректный указатель на контекст.

Примеры использования:

```

gis_core_connection_t      connection;
gis_core_request_parameters_t map;
gis_data_raw_context_t     raw_ctx;

gis_core_link_init( &connection );
gis_core_link_connect( &connection, 777 );
gis_core_databuffer_attach( &connection );
gis_core_request_parameters_init( &map );
gis_core_databuffer_data_request( &connection, &map, 1 );

gis_data_raw_ctx_alloc( &raw_ctx );
gis_data_raw_ctx_import_databuffer( raw_ctx );

    gis_core_map_list_t map_list;
gis_core_map_list_init( &map_list );
gis_core_request_maps_list( &connection, &map, GIS_CORE_DRIVERS_SXF_LOCAL, &map_list );

gis_data_raw_select_map( raw_ctx, GIS_CORE_DRIVERS_SXF_LOCAL, map_list.entry_list[0].id
↪ );

int count = gis_data_raw_map_get_object_count( raw_ctx );
if ( count != 8589 )
{
    printf( "Incorrect number of objects: %d\n", count );
    gis_data_raw_ctx_free( &raw_ctx );
        gis_core_map_list_free( &map_list );
        gis_core_databuffer_detach();
        gis_core_link_destroy( &connection, true );
    return 1;
}

gis_data_raw_ctx_free( &raw_ctx );
    gis_core_map_list_free( &map_list );
    gis_core_databuffer_detach();
gis_core_link_destroy( &connection, true );

```

Классификация:

ПК ЦКИ для ЗОСРВ «Нейтрино»

Безопасность использования	
Обработчик прерываний	Нет
Обработчик сигналов	Нет
В потоке	Нет

Тематические ссылки:

[gis_data_raw_context_t](#),

gis_data_raw_map_get_projection()

Получение проекции текущей карты в разделяемой области памяти

Прототип:

```
#include <gis/gis_raw.h>

int gis_data_raw_map_get_projection( const gis_data_raw_context_t ctx, gis_map_projection_t
↪ *pProjection );
```

Аргументы:

ctx

Контекст `gis_data_raw_context_t`.

pProjection

Указатель на поле проекции `gis_map_projection_t`.

Библиотека:

gishelper

Описание:

Функция возвращает проекцию текущей выбранной карты в разделяемой области памяти.

Возвращаемое значение:

EOK

Успешное завершение.

EINVAL

Некорректный аргумент.

EFAULT

Не удалось получить проекцию карты.

Примеры использования:

```

gis_core_connection_t      connection;
gis_core_request_parameters_t  map;
gis_data_raw_context_t     raw_ctx;

gis_core_link_init( &connection );
gis_core_link_connect( &connection, 777 );
gis_core_databuffer_attach( &connection );
gis_core_request_parameters_init( &map );
gis_core_databuffer_data_request( &connection, &map, 1 );

gis_data_raw_ctx_alloc( &raw_ctx );
gis_data_raw_ctx_import_databuffer( raw_ctx );
gis_data_raw_select_first_map( raw_ctx );

gis_map_projection_t projection;
if ( gis_data_raw_map_get_projection( raw_ctx, &projection ) != EOK ) {

    gis_data_raw_ctx_free( &raw_ctx );
    gis_core_databuffer_detach();
    gis_core_link_destroy( &connection, true );
    return -1;
}

gis_data_raw_ctx_free( &raw_ctx );
gis_core_databuffer_detach();
gis_core_link_destroy( &connection, true );

```

Классификация:

ПК ЦКИ для ЗОСРВ «Нейтрино»

Безопасность использования	
Обработчик прерываний	Нет
Обработчик сигналов	Нет
В потоке	Нет

Тематические ссылки:

[gis_data_raw_context_t](#), [gis_data_raw_select_first_map\(\)](#)

Поддерживается, начиная с ПК ЦКИ редакции 1.1.0

gis_data_raw_map_select_first_object()

Выбор первого объекта в текущей карте разделяемой области памяти.

Прототип:

```
#include <gis/gis_raw.h>

int gis_data_raw_map_select_first_object( gis_data_raw_context_t ctx );
```

Аргументы:

ctx

Контекст `gis_data_raw_context_t`.

Библиотека:

gishelper

Описание:

Функция выбирает первый объект в текущей карте разделяемой области памяти.

Возвращаемое значение:

EOK

Успешное завершение.

EINVAL

Некорректный указатель на контекст.

EBADFD

Некорректный указатель на разделяемую область памяти.

EFAULT

Некорректный указатель на текущую карту в разделяемой области памяти.

ENODATA

В выбранной карте отсутствуют объекты.

Примеры использования:

```

gis_core_connection_t      connection;
gis_core_request_parameters_t map;
gis_data_raw_context_t     raw_ctx;

gis_core_link_init( &connection );
gis_core_link_connect( &connection, 777 );
gis_core_databuffer_attach( &connection );
gis_core_request_parameters_init( &map );
gis_core_databuffer_data_request( &connection, &map, 1 );

gis_data_raw_ctx_alloc( &raw_ctx );
gis_data_raw_ctx_import_databuffer( raw_ctx );

gis_core_map_list_t map_list;
gis_core_map_list_init( &map_list );
gis_core_request_maps_list( &connection, &map, GIS_CORE_DRIVERS_SXF_LOCAL, &map_list );

gis_data_raw_select_map( raw_ctx, GIS_CORE_DRIVERS_SXF_LOCAL, map_list.entry_list[0].id
↪ );

gis_core_class_code_t cc;

if ( gis_data_raw_map_select_first_object( raw_ctx ) != EOK ) {
    printf( "gis_data_raw_map_select_first_object() failed\n" );
    gis_data_raw_ctx_free( &raw_ctx );
        gis_core_map_list_free( &map_list );
        gis_core_databuffer_detach();
        gis_core_link_destroy( &connection, true );
    return 1;
}
else
{
    cc = gis_data_raw_object_get_class_code( raw_ctx );
    if ( (int)cc != 13111000 )
    {
        printf( "Incorrect class code of first object: %d\n", (int)cc );
        gis_data_raw_ctx_free( &raw_ctx );
            gis_core_map_list_free( &map_list );
            gis_core_databuffer_detach();
            gis_core_link_destroy( &connection, true );
        return 1;
    }
}

gis_data_raw_ctx_free( &raw_ctx );
    gis_core_map_list_free( &map_list );
    gis_core_databuffer_detach();
gis_core_link_destroy( &connection, true );

```

Классификация:

ПК ЦКИ для ЗОСРВ «Нейтрино»

Безопасность использования	
Обработчик прерываний	Нет
Обработчик сигналов	Нет
В потоке	Нет

Тематические ссылки:

`gis_data_raw_context_t`, `gis_data_raw_map_select_next_object()`

gis_data_raw_map_select_next_object()

Выбор следующего объекта в текущей карте разделяемой области памяти.

Прототип:

```
#include <gis/gis_raw.h>

int gis_data_raw_map_select_next_object( gis_data_raw_context_t ctx );
```

Аргументы:

ctx

Контекст `gis_data_raw_context_t`.

Библиотека:

gishelper

Описание:

Функция выбирает следующий объект в текущей карте разделяемой области памяти.

Возвращаемое значение:

EOK

Успешное завершение.

EINVAL

Некорректный указатель на контекст.

EBADFD

Некорректный указатель на разделяемую область памяти.

EFAULT

Некорректный указатель на текущую карту в разделяемой области памяти.

EIO

Некорректный указатель на текущий объект

EDOM

Индекс объекта превышает количество объектов в карте

EAGAIN

Достигнут последний объект в карте.

Примеры использования:

```

gis_core_connection_t      connection;
gis_core_request_parameters_t map;
gis_data_raw_context_t     raw_ctx;

gis_core_link_init( &connection );
gis_core_link_connect( &connection, 777 );
gis_core_databuffer_attach( &connection );
gis_core_request_parameters_init( &map );
gis_core_databuffer_data_request( &connection, &map, 1 );

gis_data_raw_ctx_alloc( &raw_ctx );
gis_data_raw_ctx_import_databuffer( raw_ctx );

    gis_data_raw_select_first_map( raw_ctx );

    int counter = 0;
    do {
        if ( ! GIS_IS_CORE_MAP_DATA_SOURCE_VECTOR( gis_data_raw_map_get_data_source( raw_ctx
↪ ) ) )
            continue;

        if ( gis_data_raw_map_select_first_object( raw_ctx ) != EOK ) {
            printf( "gis_data_raw_map_select_first_object() failed\n" );
            gis_data_raw_ctx_free( &raw_ctx );
            gis_core_databuffer_detach();
            gis_core_link_destroy( &connection, true );
            return 1;
        }

        do {
            counter++;
        } while ( gis_data_raw_map_select_next_object( raw_ctx ) == EOK );

    } while ( gis_data_raw_select_next_map( raw_ctx ) == EOK );

    if ( counter != 29391 )
    {
        printf( "Incorrect number of objects: %d\n", counter );
        gis_data_raw_ctx_free( &raw_ctx );
        gis_core_databuffer_detach();
        gis_core_link_destroy( &connection, true );
        return 1;
    }

    gis_data_raw_ctx_free( &raw_ctx );
    gis_core_databuffer_detach();
    gis_core_link_destroy( &connection, true );

```

Классификация:

ПК ЦКИ для ЗОСРВ «Нейтрино»

Безопасность использования	
Обработчик прерываний	Нет
Обработчик сигналов	Нет
В потоке	Нет

Тематические ссылки:

`gis_data_raw_context_t`, `gis_data_raw_map_select_object()`

gis_data_raw_map_select_object()

Выбор объекта в текущей карте разделяемой области памяти по индексу.

Прототип:

```
#include <gis/gis_raw.h>

int gis_data_raw_map_select_object( gis_data_raw_context_t ctx, uint32_t index );
```

Аргументы:

ctx

Контекст `gis_data_raw_context_t`.

index

Последовательный индекс объекта в карте.

Библиотека:

gishelper

Описание:

Функция выбирает объект в текущей карте разделяемой области памяти по указанному индексу.

Возвращаемое значение:

EOK

Успешное завершение.

EINVAL

Некорректный указатель на контекст.

EBADFD

Некорректный указатель на разделяемую область памяти.

EFAULT

Некорректный указатель на текущую карту в разделяемой области памяти.

ENXIO

Некорректный указатель на текущий объект

EDOM

Не выбран текущий объект

EIO

Индекс объекта превышает количество объектов в карте

EAGAIN

Достигнут последний объект в карте.

ENOTSUP

Функция не поддерживается в ПК ЦКИ, начиная с версии 1.2 включительно. См. [gis_data_raw_map_select_n](#)

Примеры использования:

```
gis_core_connection_t      connection;
gis_core_request_parameters_t  map;
gis_data_raw_context_t      raw_ctx;

gis_core_link_init( &connection );
gis_core_link_connect( &connection, 777 );
gis_core_databuffer_attach( &connection );
gis_core_request_parameters_init( &map );
gis_core_databuffer_data_request( &connection, &map, 1 );

gis_data_raw_ctx_alloc( &raw_ctx );
gis_data_raw_ctx_import_databuffer( raw_ctx );

    gis_core_map_list_t map_list;
gis_core_map_list_init( &map_list );
gis_core_request_maps_list( &connection, &map, GIS_CORE_DRIVERS_SXF_LOCAL, &map_list );

gis_data_raw_select_map( raw_ctx, GIS_CORE_DRIVERS_SXF_LOCAL, map_list.entry_list[0].id
↵ );

if ( gis_data_raw_map_select_object( raw_ctx, 12 ) != EOK ) {
    printf( "gis_data_raw_map_select_object() failed\n" );
    gis_data_raw_ctx_free( &raw_ctx );
        gis_core_map_list_free( &map_list );
        gis_core_databuffer_detach();
        gis_core_link_destroy( &connection, true );
    return 1;
}

gis_core_class_code_t cc = gis_data_raw_object_get_class_code( raw_ctx );
printf( "%d\n", (int)cc );

gis_data_raw_ctx_free( &raw_ctx );
    gis_core_map_list_free( &map_list );
    gis_core_databuffer_detach();
gis_core_link_destroy( &connection, true );
```

Классификация:

Безопасность использования	
Обработчик прерываний	Нет
Обработчик сигналов	Нет
В потоке	Нет

Считается устаревшим, начиная с ПК ЦКИ редакции 1.2

Тематические ссылки:

`gis_data_raw_context_t, gis_data_raw_map_select_next_object()`

gis_data_raw_object_get_bounding_rect()

Получение описывающего прямоугольника для объекта.

Прототип:

```
#include <gis/gis_raw.h>

int gis_data_raw_object_get_bounding_rect( gis_data_raw_context_t ctx, gis_borders_t
↵ *bounding_rect );
```

Аргументы:

ctx

Контекст `gis_data_raw_context_t`.

bounding_rect

Указатель на структуру `gis_borders_t`, куда будут сохранены границы описывающего прямоугольника.

Библиотека:

gishelper

Описание:

Функция получает градусные границы описывающего прямоугольника для объекта.

Возвращаемое значение:

EOK

Успешное завершение.

EFAULT

Некорректный указатель на контекст или структуру.

ENODATA

Не удалось получить объект.

Примеры использования:

```

gis_object_t object;
gis_data_raw_context_t raw_ctx;

gis_object_init( &object );

gis_core_connection_t      connection;
gis_core_request_parameters_t map;

gis_core_link_init( &connection );
gis_core_link_connect( &connection, 777 );
gis_core_databuffer_attach( &connection );
gis_core_request_parameters_init( &map );
gis_core_databuffer_data_request( &connection, &map, 1 );

gis_data_raw_ctx_alloc( &raw_ctx );
gis_data_raw_ctx_import_databuffer( raw_ctx );

    gis_core_map_list_t map_list;
gis_core_map_list_init( &map_list );
gis_core_request_maps_list( &connection, &map, GIS_CORE_DRIVERS_SXF_LOCAL, &map_list );

gis_data_raw_select_map( raw_ctx, GIS_CORE_DRIVERS_SXF_LOCAL, map_list.entry_list[0].id
↪ );

gis_data_raw_map_select_first_object( raw_ctx );

gis_data_raw_object_get_bounding_rect( raw_ctx, &object.bounding_rect );

if ( fabs ( object.bounding_rect.degrees.west  - 39.997025 < 0.0001 ) &&
    fabs ( object.bounding_rect.degrees.east  - 39.998626 < 0.0001 ) &&
    fabs ( object.bounding_rect.degrees.north - 43.506432 < 0.0001 ) &&
    fabs ( object.bounding_rect.degrees.south - 43.500663 < 0.0001 ) )
{
    printf("Correct\n" );
}
else
{
    printf("Incorrect degree coordinates: %lf %lf %lf %lf\n",
↪ object.bounding_rect.degrees.west,  object.bounding_rect.degrees.east,
↪ object.bounding_rect.degrees.north, object.bounding_rect.degrees.south );
    gis_data_raw_ctx_free( &raw_ctx );
    gis_object_free( &object );
    gis_core_map_list_free( &map_list );
    gis_core_databuffer_detach();
    gis_core_link_destroy( &connection, true );
    return 1;
}

gis_data_raw_ctx_free( &raw_ctx );

```

```
gis_object_free( &object );
    gis_core_map_list_free( &map_list );
    gis_core_databuffer_detach();
gis_core_link_destroy( &connection, true );
```

Классификация:

ПК ЦКИ для ЗОСРВ «Нейтрино»

Безопасность использования	
Обработчик прерываний	Нет
Обработчик сигналов	Нет
В потоке	Нет

Поддерживается, начиная с ПК ЦКИ редакции 1.1.0

Тематические ссылки:

gis_data_raw_context_t, gis_borders_t

gis_data_raw_object_get_class_code()

Получение кода класса выбранного объекта текущей карты.

Прототип:

```
#include <gis/gis_raw.h>

gis_core_class_code_t gis_data_raw_object_get_class_code( gis_data_raw_context_t ctx );
```

Аргументы:

ctx

Контекст `gis_data_raw_context_t`.

Библиотека:

gishelper

Описание:

Функция возвращает код класса выбранного объекта текущей карты.

Возвращаемое значение:

`!GIS_CLASS_CODE_UNDEFINED`
Успешное завершение.

`GIS_CLASS_CODE_UNDEFINED`
Не удалось получить код класса выбранного объекта.

Примеры использования:

```
gis_core_connection_t      connection;
gis_core_request_parameters_t map;
gis_data_raw_context_t     raw_ctx;

gis_core_link_init( &connection );
gis_core_link_connect( &connection, 777 );
gis_core_databuffer_attach( &connection );
gis_core_request_parameters_init( &map );
gis_core_databuffer_data_request( &connection, &map, 1 );

gis_data_raw_ctx_alloc( &raw_ctx );
```

```

gis_data_raw_ctx_import_databuffer( raw_ctx );

    gis_core_map_list_t map_list;
    gis_core_map_list_init( &map_list );
    gis_core_request_maps_list( &connection, &map, GIS_CORE_DRIVERS_SXF_LOCAL, &map_list );

    gis_data_raw_select_map( raw_ctx, GIS_CORE_DRIVERS_SXF_LOCAL, map_list.entry_list[0].id
↵ );

    if ( gis_data_raw_map_select_first_object( raw_ctx ) != EOK )
    {
        printf( "gis_data_raw_map_select_first_object() failed\n" );
        gis_data_raw_ctx_free( &raw_ctx );
        gis_core_map_list_free( &map_list );
        gis_core_databuffer_detach();
        gis_core_link_destroy( &connection, true );
        return 1;
    }

    gis_core_class_code_t code = gis_data_raw_object_get_class_code( raw_ctx );

    if ( (int)code != 13111000 )
    {
        printf( "Incorrect code %d\n", (int)code );
        gis_data_raw_ctx_free( &raw_ctx );
        gis_core_map_list_free( &map_list );
        gis_core_databuffer_detach();
        gis_core_link_destroy( &connection, true );
        return 1;
    }

    gis_data_raw_ctx_free( &raw_ctx );
    gis_core_map_list_free( &map_list );
    gis_core_databuffer_detach();
    gis_core_link_destroy( &connection, true );

```

Классификация:

ПК ЦКИ для ЗОСРВ «Нейтрино»

Безопасность использования	
Обработчик прерываний	Нет
Обработчик сигналов	Нет
В потоке	Нет

Тематические ссылки:

[gis_data_raw_context_t](#)

gis_data_raw_object_get_data()

Получение массива точек выбранного объекта текущей карты.

Прототип:

```
#include <gis/gis_raw.h>

object_point_t * gis_data_raw_object_get_data( gis_data_raw_context_t ctx );
```

Аргументы:

ctx

Контекст `gis_data_raw_context_t`.

Библиотека:

gishelper

Описание:

Функция возвращает указатель на массив точек выбранного объекта текущей карты.

Возвращаемое значение:

!NULL

Успешное завершение.

NULL

Не удалось получить массив точек выбранного объекта.

Примеры использования:

```
gis_core_connection_t      connection;
gis_core_request_parameters_t map;
gis_data_raw_context_t     raw_ctx;

gis_core_link_init( &connection );
gis_core_link_connect( &connection, 777 );
gis_core_databuffer_attach( &connection );
gis_core_request_parameters_init( &map );
gis_core_databuffer_data_request( &connection, &map, 1 );

gis_data_raw_ctx_alloc( &raw_ctx );
```

```

gis_data_raw_ctx_import_databuffer( raw_ctx );

gis_core_map_list_t map_list;
gis_core_map_list_init( &map_list );
gis_core_request_maps_list( &connection, &map, GIS_CORE_DRIVERS_SXF_LOCAL, &map_list );

gis_data_raw_select_map( raw_ctx, GIS_CORE_DRIVERS_SXF_LOCAL, map_list.entry_list[0].id
↪ );

gis_data_raw_map_select_first_object( raw_ctx );

object_point_t *points = gis_data_raw_object_get_data( raw_ctx );

if ( fabs( points->degrees.x - 39.015295 ) < 0.0001 &&
    fabs( points->degrees.y - 43.333212 ) < 0.0001 &&
    fabs( points->meters.x - 7501351.669930 ) < 0.0001 &&
    fabs( points->meters.y - 4799838.621442 ) < 0.0001 )
{
    printf( "Correct\n" );
}
else
{
    printf( "Incorrect data values\n" );
    printf( " degrees.x = %f degrees.y = %f\n meters.x = %f meters.y = %f\n",
        points->degrees.x, points->degrees.y,
↪ points->meters.x, points->meters.y );
    gis_data_raw_ctx_free( &raw_ctx );
    gis_core_map_list_free( &map_list );
    gis_core_databuffer_detach();
    gis_core_link_destroy( &connection, true );
    return 1;
}

gis_data_raw_ctx_free( &raw_ctx );
gis_core_map_list_free( &map_list );
gis_core_databuffer_detach();
gis_core_link_destroy( &connection, true );

```

Классификация:

ПК ЦКИ для ЗОСРВ «Нейтрино»

Безопасность использования	
Обработчик прерываний	Нет
Обработчик сигналов	Нет
В потоке	Нет

Тематические ссылки:

`gis_data_raw_context_t, gis_data_raw_object_get_point_count()`

gis_data_raw_object_get_height()

Получение высоты выбранного объекта текущей карты.

Прототип:

```
#include <gis/gis_raw.h>

double gis_data_raw_object_get_height( gis_data_raw_context_t ctx );
```

Аргументы:

ctx

Контекст `gis_data_raw_context_t`.

Библиотека:

gishelper

Описание:

Функция возвращает высоту выбранного объекта текущей карты.

Возвращаемое значение:

!0xFFFFFFFF

Успешное завершение.

0xFFFFFFFF

Не удалось получить высоту выбранного объекта.

Примеры использования:

```
gis_core_connection_t      connection;
gis_core_request_parameters_t map;
gis_data_raw_context_t     raw_ctx;

gis_core_link_init( &connection );
gis_core_link_connect( &connection, 777 );
gis_core_databuffer_attach( &connection );
gis_core_request_parameters_init( &map );
gis_core_databuffer_data_request( &connection, &map, 1 );

gis_data_raw_ctx_alloc( &raw_ctx );
```

```

gis_data_raw_ctx_import_databuffer( raw_ctx );

    gis_core_map_list_t map_list;
gis_core_map_list_init( &map_list );
gis_core_request_maps_list( &connection, &map, GIS_CORE_DRIVERS_SXF_LOCAL, &map_list );

gis_data_raw_select_map( raw_ctx, GIS_CORE_DRIVERS_SXF_LOCAL, map_list.entry_list[0].id
↵ );

gis_data_raw_map_select_first_object( raw_ctx );

double height = gis_data_raw_object_get_height( raw_ctx );
if ( fabs( height - 0.000000 ) > 0.0001 )
{
    printf( "Incorrect height: %f!\n", height );
    gis_data_raw_ctx_free( &raw_ctx );
        gis_core_map_list_free( &map_list );
    gis_core_databuffer_detach();
        gis_core_link_destroy( &connection, true );
    return 1;
}

gis_data_raw_ctx_free( &raw_ctx );
    gis_core_map_list_free( &map_list );
    gis_core_databuffer_detach();
gis_core_link_destroy( &connection, true );

```

Классификация:

ПК ЦКИ для ЗОСРВ «Нейтрино»

Безопасность использования	
Обработчик прерываний	Нет
Обработчик сигналов	Нет
В потоке	Нет

Тематические ссылки:

[gis_data_raw_context_t](#), [gis_data_raw_object_get_data\(\)](#)

gis_data_raw_object_get_point_count()

Получение количества точек в выбранном объекте текущей карты.

Прототип:

```
#include <gis/gis_raw.h>

int32_t gis_data_raw_object_get_point_count( gis_data_raw_context_t ctx );
```

Аргументы:

ctx

Контекст `gis_data_raw_context_t`.

Библиотека:

gishelper

Описание:

Функция возвращает количество точек в выбранном объекте текущей карты.

Возвращаемое значение:

>=0

Успешное завершение.

-1

Не удалось получить массив точек выбранного объекта.

Примеры использования:

```
gis_core_connection_t      connection;
gis_core_request_parameters_t map;
gis_data_raw_context_t     raw_ctx;

gis_core_link_init( &connection );
gis_core_link_connect( &connection, 777 );
gis_core_databuffer_attach( &connection );
gis_core_request_parameters_init( &map );
gis_core_databuffer_data_request( &connection, &map, 1 );

gis_data_raw_ctx_alloc( &raw_ctx );
```

```

gis_data_raw_ctx_import_databuffer( raw_ctx );

    gis_core_map_list_t map_list;
    gis_core_map_list_init( &map_list );
    gis_core_request_maps_list( &connection, &map, GIS_CORE_DRIVERS_SXF_LOCAL, &map_list );

    gis_data_raw_select_map( raw_ctx, GIS_CORE_DRIVERS_SXF_LOCAL, map_list.entry_list[0].id
↵ );

    gis_data_raw_map_select_first_object( raw_ctx );

    int32_t count = gis_data_raw_object_get_point_count( raw_ctx );
    if ( count != 2 )
    {
        printf( "Incorrect point count for selected object: %d.\n", count);
        gis_data_raw_ctx_free( &raw_ctx );
        gis_core_map_list_free( &map_list );
        gis_core_databuffer_detach();
        gis_core_link_destroy( &connection, true );
        return 1;
    }

    gis_data_raw_ctx_free( &raw_ctx );
    gis_core_map_list_free( &map_list );
    gis_core_databuffer_detach();
    gis_core_link_destroy( &connection, true );

```

Классификация:

ПК ЦКИ для ЗОСРВ «Нейтрино»

Безопасность использования	
Обработчик прерываний	Нет
Обработчик сигналов	Нет
В потоке	Нет

Тематические ссылки:

[gis_data_raw_context_t](#), [gis_data_raw_object_get_data\(\)](#)

gis_data_raw_object_get_type()

Получение типа примитива выбранного объекта текущей карты.

Прототип:

```
#include <gis/gis_raw.h>

gis_object_primitive_type_t gis_data_raw_object_get_type( gis_data_raw_context_t ctx );
```

Аргументы:

ctx

Контекст `gis_data_raw_context_t`.

Библиотека:

gishelper

Описание:

Функция возвращает тип примитива выбранного объекта текущей карты.

Возвращаемое значение:

`!GIS_OBJECT_PRIMITIVE_UNKNOWN`

Успешное завершение.

`GIS_OBJECT_PRIMITIVE_UNKNOWN`

Не удалось получить тип примитива выбранного объекта.

Примеры использования:

```
gis_core_connection_t      connection;
gis_core_request_parameters_t map;
gis_data_raw_context_t     raw_ctx;

gis_core_link_init( &connection );
gis_core_link_connect( &connection, 777 );
gis_core_databuffer_attach( &connection );
gis_core_request_parameters_init( &map );
gis_core_databuffer_data_request( &connection, &map, 1 );

gis_data_raw_ctx_alloc( &raw_ctx );
```

```

gis_data_raw_ctx_import_databuffer( raw_ctx );

    gis_core_map_list_t map_list;
    gis_core_map_list_init( &map_list );
    gis_core_request_maps_list( &connection, &map, GIS_CORE_DRIVERS_SXF_LOCAL, &map_list );

    gis_data_raw_select_map( raw_ctx, GIS_CORE_DRIVERS_SXF_LOCAL, map_list.entry_list[0].id
↪ );

    if ( gis_data_raw_map_select_first_object( raw_ctx ) != EOK )
    {
        printf( "gis_data_raw_map_select_first_object() failed\n" );
        gis_data_raw_ctx_free( &raw_ctx );
        gis_core_map_list_free( &map_list );
        gis_core_databuffer_detach();
        gis_core_link_destroy( &connection, true );
        return 1;
    }

    gis_object_primitive_type_t type = gis_data_raw_object_get_type( raw_ctx );
    if ( type != GIS_OBJECT_PRIMITIVE_LINE )
    {
        printf( "Incorrect object type: %d, expected: %d\n",
↪
type,
↪ GIS_OBJECT_PRIMITIVE_LINE );
        gis_data_raw_ctx_free( &raw_ctx );
        gis_core_map_list_free( &map_list );
        gis_core_databuffer_detach();
        gis_core_link_destroy( &connection, true );
        return 1;
    }

    gis_data_raw_ctx_free( &raw_ctx );
    gis_core_map_list_free( &map_list );
    gis_core_databuffer_detach();
    gis_core_link_destroy( &connection, true );

```

Классификация:

ПК ЦКИ для ЗОСРВ «Нейтрино»

Безопасность использования	
Обработчик прерываний	Нет
Обработчик сигналов	Нет
В потоке	Нет

Тематические ссылки:

`gis_data_raw_context_t`, `gis_object_primitive_type_t`, `gis_data_raw_object_get_data()`

gis_data_raw_raster_maps_get_height_limits()

Получение пределов высот для всех растровых карт в разделяемой области памяти.

Прототип:

```
#include <gis/gis_raw.h>

int gis_data_raw_raster_maps_get_height_limits( gis_data_raw_context_t ctx, double_point_t
↵ *limits );
```

Аргументы:

ctx

Контекст `gis_data_raw_context_t`.

limits

Указатель на `int32_point_t`, `float_point_t`, `double_point_t` (.x - нижний предел, .y - верхний предел).

Библиотека:

gishelper

Описание:

Функция заполняет пределы высот для всех растровых карт в разделяемой области памяти.

Возвращаемое значение:

EOK

Успешное завершение.

EINVAL

Некорректный указатель на контекст.

EFAULT

Не выбрана карта.

ENODATA

В разделяемой области памяти отсутствуют растровые карты.

Примеры использования:

```

gis_core_connection_t      connection;
gis_core_request_parameters_t map;
gis_data_raw_context_t     raw_ctx;

gis_core_link_init( &connection );
gis_core_link_connect( &connection, 777 );
gis_core_databuffer_attach( &connection );
gis_core_request_parameters_init( &map );
gis_core_databuffer_data_request( &connection, &map, 1 );

gis_data_raw_ctx_alloc( &raw_ctx );
gis_data_raw_ctx_import_databuffer( raw_ctx );

double_point_t limits;

int result = gis_data_raw_raster_maps_get_height_limits( raw_ctx, &limits );
if ( result == ENODATA ) {
    printf( "No raster maps\n" );
    gis_data_raw_ctx_free( &raw_ctx );
    gis_core_databuffer_detach();
    gis_core_link_destroy( &connection, true );
    return 1;
} else if ( result != EOK ) {
    printf( "Failed to get raster limits\n" );
    gis_data_raw_ctx_free( &raw_ctx );
    gis_core_databuffer_detach();
    gis_core_link_destroy( &connection, true );
    return 1;
}

if ( fabs( limits.x - 75.500000 ) < 0.0001 &&
    fabs( limits.y - 3983.000000 ) < 0.0001 )
{
    printf( "Correct\n" );
}
else
{
    printf( "Incorrect height limits\n" );
    gis_data_raw_ctx_free( &raw_ctx );
    gis_core_databuffer_detach();
    gis_core_link_destroy( &connection, true );
    return 1;
}

gis_data_raw_ctx_free( &raw_ctx );
gis_core_databuffer_detach();
gis_core_link_destroy( &connection, true );

```

Безопасность использования	
Обработчик прерываний	Нет
Обработчик сигналов	Нет
В потоке	Нет

Тематические ссылки:

[gis_data_raw_context_t](#), [gis_data_raw_object_get_height\(\)](#)

gis_data_raw_select_first_map()

Выбор первой карты в разделяемой области памяти.

Прототип:

```
#include <gis/gis_raw.h>

int gis_data_raw_select_first_map( gis_data_raw_context_t ctx );
```

Аргументы:

ctx

Контекст `gis_data_raw_context_t`.

Библиотека:

gishelper

Описание:

Функция выбирает первую карту в разделяемой области памяти.

Возвращаемое значение:

EOK

Успешное завершение.

EBADFD

Некорректный указатель на разделяемую область памяти.

EINVAL

Некорректный контекст.

ENODATA

Отсутствуют карты в разделяемой области памяти.

Примеры использования:

```

gis_core_connection_t      connection;
gis_core_request_parameters_t  map;
gis_data_raw_context_t     raw_ctx;

gis_core_link_init( &connection );
gis_core_link_connect( &connection, 777 );
gis_core_databuffer_attach( &connection );
gis_core_request_parameters_init( &map );
gis_core_databuffer_data_request( &connection, &map, 1 );

gis_data_raw_ctx_alloc( &raw_ctx );
gis_data_raw_ctx_import_databuffer( raw_ctx );

if ( gis_data_raw_select_first_map( raw_ctx ) != EOK )
{
    printf( "Failed to select first map!\n" );
    gis_data_raw_ctx_free( &raw_ctx );
        gis_core_databuffer_detach();
        gis_core_link_destroy( &connection, true );
    return 1;
}

gis_data_raw_ctx_free( &raw_ctx );
    gis_core_databuffer_detach();
gis_core_link_destroy( &connection, true );

```

Классификация:

ПК ЦКИ для ЗОСРВ «Нейтрино»

Безопасность использования	
Обработчик прерываний	Нет
Обработчик сигналов	Нет
В потоке	Нет

Тематические ссылки:

[gis_data_raw_context_t](#), [gis_data_raw_select_next_map\(\)](#)

gis_data_raw_select_map()

Выбор требуемой карты в разделяемой области памяти.

Прототип:

```
#include <gis/gis_raw.h>

int gis_data_raw_select_map( gis_data_raw_context_t    ctx,
                             gis_core_driver_id_t     driver_id,
                             int32_t                  map_id );
```

Аргументы:

ctx

Контекст `gis_data_raw_context_t`.

driver_id

Идентификатор драйвера требуемой карты (`gis_core_driver_id_t`).

map_id

Индекс карты в рамках выбранного драйвера.

Библиотека:

gishelper

Описание:

Функция выбирает карту с переданным порядковым номером (индексом) `map_id`, которая принадлежит указанному идентификатору драйвера `driver_id`. Все карты в рамках одного драйвера имеют уникальный численный идентификатор. Таким образом, указав связку `driver_id` и `map_id` можно однозначно определить конкретную карту. Для получения списка карт заданного драйвера можно воспользоваться функцией `gis_core_request_maps_list()`. Функция вернет `gis_core_map_list_t`, пройдя по которому в цикле можно получить идентификатор интересующей карты, и затем передать этот идентификатор в `gis_data_raw_select_map()` для низкоуровневой работы с данными карты.

Возвращаемое значение:

EOK

Успешное завершение.

EFAULT

Некорректный идентификатор драйвера.

EINVAL

Некорректный индекс карты.

EBADFD

Некорректное обращение к разделяемой области памяти.

ENODATA

Отсутствуют карты в разделяемой области памяти.

Примеры использования:

```
gis_core_connection_t      connection;
gis_core_request_parameters_t  params;
gis_data_raw_context_t     raw_ctx;

gis_core_link_init( &connection );
gis_core_link_connect( &connection, 777 );
gis_core_databuffer_attach( &connection );
gis_core_request_parameters_init( &params );
gis_core_databuffer_data_request( &connection, &params, 1 );

gis_data_raw_ctx_alloc( &raw_ctx );
gis_data_raw_ctx_import_databuffer( raw_ctx );

gis_core_map_list_t map_list;
gis_core_map_list_init( &map_list );
gis_core_request_maps_list( &connection, &params, GIS_CORE_DRIVERS_SXF_LOCAL, &map_list
↪ );

gis_data_raw_select_map( raw_ctx, GIS_CORE_DRIVERS_SXF_LOCAL, map_list.entry_list[0].id
↪ );
int count_obj = gis_data_raw_map_get_object_count( raw_ctx );
if ( count_obj != 8589 )
{
    printf( "Incorrect number of objects: %d\n", count_obj );
    gis_data_raw_ctx_free( &raw_ctx );
    gis_core_map_list_free( &map_list );
    gis_core_databuffer_detach();
    gis_core_link_destroy( &connection, true );
    return 1;
}

gis_data_raw_ctx_free( &raw_ctx );
gis_core_map_list_free( &map_list );
gis_core_databuffer_detach();
gis_core_link_destroy( &connection, true );
```

Классификация:

ПК ЦКИ для ЗОСРВ «Нейтрино»

Безопасность использования	
<i>Обработчик прерываний</i>	Нет
<i>Обработчик сигналов</i>	Нет
<i>В потоке</i>	Нет

Поддерживается, начиная с ПК ЦКИ редакции 1.1.0

Тематические ссылки:

`gis_data_raw_context_t`, `gis_data_raw_select_first_map()`, `gis_data_raw_select_next_map()`

gis_data_raw_select_map_by_index()

Выбор карты в разделяемой области памяти по индексу.

Прототип:

```
#include <gis/gis_raw.h>

int gis_data_raw_select_map_by_index( gis_data_raw_context_t ctx, uint32_t index );
```

Аргументы:

ctx

Контекст `gis_data_raw_context_t`.

index

Индекс карты.

Библиотека:

gishelper

Описание:

Функция выбирает карту с переданным порядковым номером (индексом) в разделяемой области памяти. Функция выбирает только активные карты, попытка выбрать неактивную карту приведет к ошибке.



При обновлении/перезаписи РОП карты могут изменить свое положение в РОП и, как следствие, изменить свой порядковый номер. Для обхода карт в РОП рекомендуется использовать `gis_data_raw_select_first_map()` и `gis_data_raw_select_next_map()`.

Возвращаемое значение:

EOK

Успешное завершение.

EINVAL

Некорректный индекс.

EBADFD

Некорректный указатель на разделяемую область памяти.

ENODATA

Отсутствуют карты в разделяемой области памяти.

Примеры использования:

```
gis_core_connection_t      connection;
gis_core_request_parameters_t  map;
gis_data_raw_context_t     raw_ctx;

gis_core_link_init( &connection );
gis_core_link_connect( &connection, 777 );
gis_core_databuffer_attach( &connection );
gis_core_request_parameters_init( &map );
gis_core_databuffer_data_request( &connection, &map, 1 );

gis_data_raw_ctx_alloc( &raw_ctx );
gis_data_raw_ctx_import_databuffer( raw_ctx );

if ( gis_data_raw_select_map_by_index( raw_ctx, 0 ) != EOK )
{
    printf( "Failed to select map by index: %d!\n", 0 );
    gis_data_raw_ctx_free( &raw_ctx );
    gis_core_databuffer_detach();
    gis_core_link_destroy( &connection, true );
    return 1;
}

gis_data_raw_ctx_free( &raw_ctx );
gis_core_databuffer_detach();
gis_core_link_destroy( &connection, true );
```

Классификация:

ПК ЦКИ для ЗОСРВ «Нейтрино»

Безопасность использования	
Обработчик прерываний	Нет
Обработчик сигналов	Нет
В потоке	Нет

Считается устаревшим, начиная с ПК ЦКИ редакции 1.1.0

Тематические ссылки:

[gis_data_raw_context_t](#), [gis_data_raw_select_next_map\(\)](#)

gis_data_raw_select_next_map()

Выбор следующей карты в разделяемой области памяти.

Прототип:

```
#include <gis/gis_raw.h>

int gis_data_raw_select_next_map( gis_data_raw_context_t ctx );
```

Аргументы:

ctx

Контекст `gis_data_raw_context_t`.

Библиотека:

gishelper

Описание:

Функция выбирает следующую карту в разделяемой области памяти.

Возвращаемое значение:

EOK

Успешное завершение.

EINVAL

Некорректный указатель на контекст.

EBADFD

Некорректный указатель на разделяемую область памяти.

EDOM

Некорректный указатель на текущую карту в разделяемой области памяти. Убедитесь в корректности вызова `gis_data_raw_select_first_map()`

EAGAIN

Карты в разделяемой области памяти закончились.

Примеры использования:

```

gis_core_connection_t      connection;
gis_core_request_parameters_t  map;
gis_data_raw_context_t     raw_ctx;

gis_core_link_init( &connection );
gis_core_link_connect( &connection, 777 );
gis_core_databuffer_attach( &connection );
gis_core_request_parameters_init( &map );
gis_core_databuffer_data_request( &connection, &map, 1 );

gis_data_raw_ctx_alloc( &raw_ctx );
gis_data_raw_ctx_import_databuffer( raw_ctx );
gis_data_raw_select_first_map( raw_ctx );

int count = 0;
do {
    count++;
} while ( gis_data_raw_select_next_map( raw_ctx ) == EOK);

if ( count != gis_data_raw_get_map_count( raw_ctx ) )
{
    printf( "Incorrect number of iterations for gis_data_raw_select_next_map(): %d\n",
↪ count );
    gis_data_raw_ctx_free( &raw_ctx );
    gis_core_databuffer_detach();
    gis_core_link_destroy( &connection, true );
    return 1;
}

gis_data_raw_ctx_free( &raw_ctx );
gis_core_databuffer_detach();
gis_core_link_destroy( &connection, true );

```

Классификация:

ПК ЦКИ для ЗОСРВ «Нейтрино»

Безопасность использования	
Обработчик прерываний	Нет
Обработчик сигналов	Нет
В потоке	Нет

Тематические ссылки:

[gis_data_raw_context_t](#), [gis_data_raw_select_first_map\(\)](#)

gis_gui_about()

Вывод информации о ПК ЦКИ.

Прототип:

```
#include <gis/gisrender.h>

int gis_gui_about( gis_core_connection_t *connection, void *parent );
```

Аргументы:

connection

Указатель на дескриптор соединения с ядром.

parent

Указатель на виджет-предок.

Библиотека:

gisrender

Описание:

Функция создаёт окно с информацией о разработчике и сведения о текущей версии продукта (ядра).

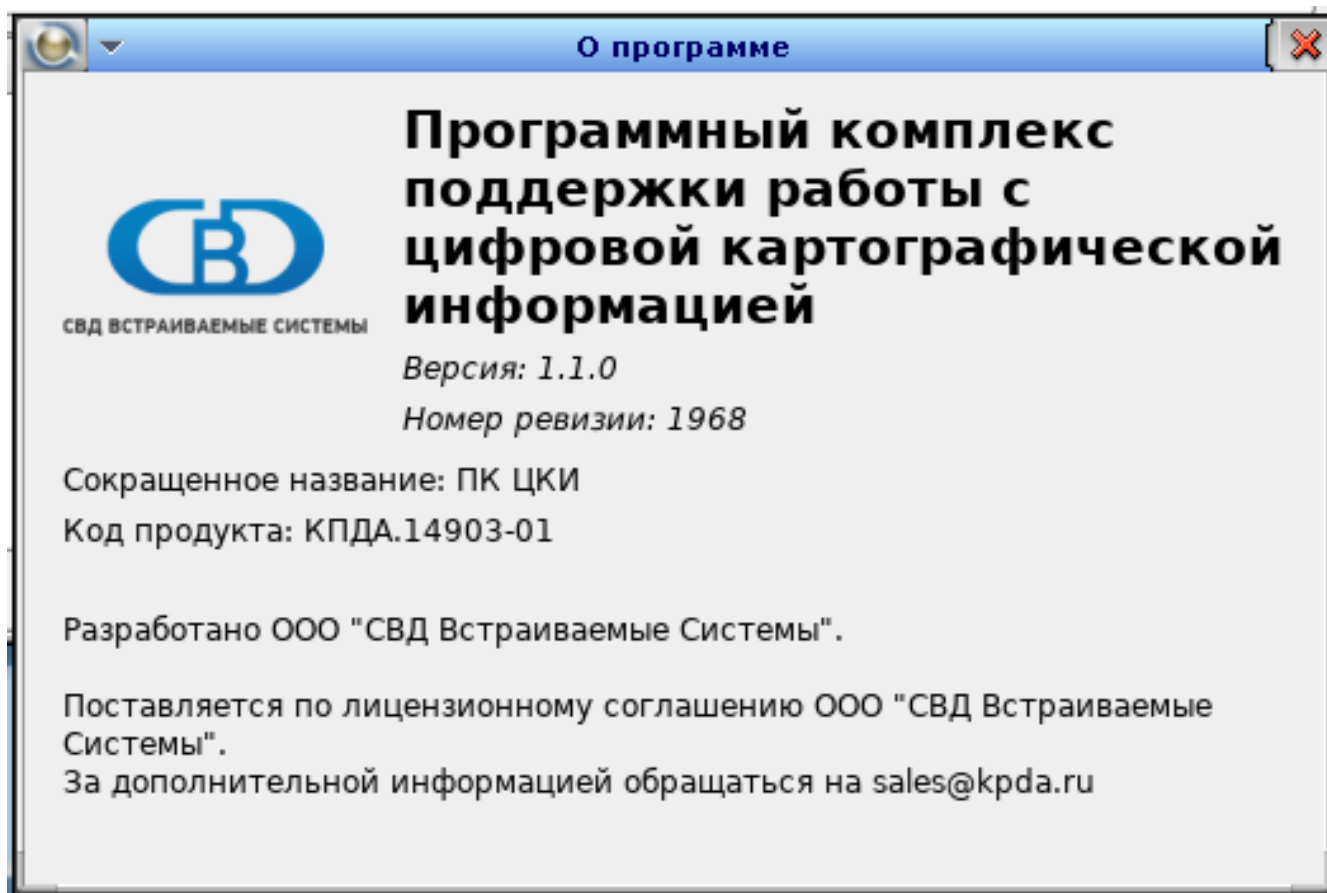


Рисунок 8. Окно просмотра информации о версии продукта (ядра).

Возвращаемое значение:

EOK

Успешное завершение

EFAULT

Не удалось отобразить окно или получить сведения о версии ядра

Примеры использования:

```
QApplication app( argc, argv );
MapWidget w;

gis_core_connection_t *connection;
gis_core_link_init( &connection );
gis_core_link_connect( &connection, 777 );

gis_gui_about( connection, &w );

gis_core_link_destroy( &connection, true );
```

Поддерживается, начиная с ПК ЦКИ редакции 1.2

Классификация:

ПК ЦКИ для ЗОСРВ «Нейтрино»

Безопасность использования	
Обработчик прерываний	Нет
Обработчик сигналов	Нет
В потоке	Да

Тематические ссылки:

`gis_gui_get_projection_parameters()` `gis_gui_get_screen_parameters()`

gis_helper_are_maps_equal()

Сравнение двух карт.

Прототип:

```
#include <gis/gishelper.h>

int gis_helper_are_maps_equal( gis_core_map_information_t *entry1,
↪ gis_core_map_information_t *entry2, int hash_check, int borders_check );
```

Аргументы:

entry1

Указатель на информацию о карте во внутреннем формате.

entry2

Указатель на информацию о карте во внутреннем формате.

hash_check

Флаг сравнения хэша карт (0 - не сравнивать хэши).

borders_check

Флаг сравнения границ карт (0 - не сравнивать границы).

Библиотека:

gishelper

Описание:

Функция позволяет сравнить две карты во внутреннем формате (GCM).

Возвращаемое значение:

0

Карты различны.

1

Карты одинаковы.

Примеры использования:

```

gis_core_connection_t    connection;
gis_core_link_init( &connection );
gis_core_link_connect( &connection, 777 );

gis_core_request_parameters_t map_desc;
gis_core_request_parameters_init( &map_desc );

gis_core_map_list_t map_list;
gis_core_map_list_init( &map_list );
gis_core_request_maps_list( &connection, &map_desc, GIS_CORE_DRIVERS_SXF_LOCAL, &map_list )
↵ ;

gis_core_map_information_t *actual_map = &map_list.entry_list[0];
gis_core_map_information_t *check_map = &map_list.entry_list[1];

if ( gis_helper_are_maps_equal( actual_map, check_map, 0, 0 ) == 1 )
{
    printf( "Maps are equal, but they should not be.\n" );
    return 1;
}

gis_core_map_list_free( &map_list );
gis_core_link_destroy( &connection, true );

```

Классификация:

ПК ЦКИ для ЗОСРВ «Нейтрино»

Безопасность использования	
Обработчик прерываний	Да
Обработчик сигналов	Да
В потоке	Да

Тематические ссылки:

[gis_core_map_information_t](#), [gis_core_map_list_t](#), [gis_core_map_list_get_entry\(\)](#),

gis_helper_convert_point_degrees_2_meters()

Преобразование координат точки (градусы) в координаты точки в метрической системе проекции.

Прототип:

```
#include <gis/gis_math.h>

int gis_helper_convert_point_degrees_2_meters( gis_helper_math_ctx_t math_ctx,
↪ double_point_t *degrees, double_point_t *out_meters );
```

Аргументы:

math_ctx

Указатель на математический контекст.

degrees

Указатель на точку с координатами в градусах, тип `int32_point_t`, `float_point_t`, `double_point_t`

out_meters

Указатель на точку, в которую будет записан результат преобразования в метрах, тип `int32_point_t`, `float_point_t`, `double_point_t`

Библиотека:

gishelper

Описание:

Функция используется для преобразования координат из градусов в метры.

Возвращаемое значение:

EOK

Успешное выполнение

EFAULT

Преобразование завершилось с ошибкой

EINVAL

Некорректный указатель

Примеры использования:

```

gis_helper_math_ctx_t math_ctx;
gis_helper_math_init_ctx( &math_ctx );

gis_map_projection_t projection;
gis_map_projection_init( &projection );
projection.EPSG = 3857;
gis_helper_math_generate_meters_projection( math_ctx, &projection );

double_point_t degree;
double_point_t meters;

degree.x = 45;
degree.y = 40;

if ( gis_helper_convert_point_degrees_2_meters( math_ctx, &degree, &meters ) != EOK ) {
    printf( "Conversion failed" );
    return 1;
}

if ( fabs( meters.x - 5009377.085697 ) < 0.00001 &&
    fabs( meters.y - 4865942.279503 ) < 0.00001 ) {
    printf( "Correct" );
} else {
    printf( "Incorrect meters coordinates: %lf %lf", meters.x, meters.y );
    return 1;
}

```

Классификация:

ПК ЦКИ для ЗОСРВ «Нейтрино»

Безопасность использования	
Обработчик прерываний	Нет
Обработчик сигналов	Нет
В потоке	Да

Тематические ссылки:

[gis_helper_math_ctx_t](#), [gis_helper_convert_point_meters_2_degrees\(\)](#)

gis_helper_convert_point_meters_2_degrees()

Преобразование координат точки в метрах (проекции) в координаты градусной меры.

Прототип:

```
#include <gis/gis_math.h>

int gis_helper_convert_point_meters_2_degrees( gis_helper_math_ctx_t math_ctx,
↪ double_point_t *out_degrees, double_point_t *meters );
```

Аргументы:

math_ctx

Указатель на математический контекст.

out_degrees

Указатель на точку, в которую будет записан результат преобразования в градусах, тип `int32_point_t`, `float_point_t`, `double_point_t`

meters

Указатель на точку с координатами в метрах, тип `int32_point_t`, `float_point_t`, `double_point_t`

Библиотека:

gishelper

Описание:

Функция используется для преобразования координат из метров в градусы.

Возвращаемое значение:

EOK

Успешное выполнение

EFAULT

Преобразование завершилось с ошибкой

EINVAL

Некорректный указатель

Примеры использования:

```

gis_helper_math_ctx_t math_ctx;
gis_helper_math_init_ctx( &math_ctx );

gis_map_projection_t projection;
gis_map_projection_init( &projection );
projection.EPSG = 3857;
gis_helper_math_generate_meters_projection( math_ctx, &projection );

double_point_t degree;
double_point_t meters;
meters.x = 1000;
meters.y = 2000;

if ( gis_helper_convert_point_meters_2_degrees( math_ctx, &degree, &meters ) != EOK ) {
    printf( "gis_helper_convert_point_meters_2_degrees() failed" );
    return 1;
}

if ( fabs( degree.x - 0.008983 ) < 0.00001 &&
    fabs( degree.y - 0.017966 ) < 0.00001 ) {
    printf( "Correct" );
} else {
    printf( "Incorrect degree coordinates: %lf %lf", degree.x, degree.y );
}

```

Классификация:

ПК ЦКИ для ЗОСРВ «Нейтрино»

Безопасность использования	
Обработчик прерываний	Нет
Обработчик сигналов	Нет
В потоке	Да

Тематические ссылки:

[int32_point_t](#), [float_point_t](#), [double_point_t](#), [gis_helper_math_ctx_t](#), [gis_helper_convert_point_degrees_2_meters\(\)](#)

gis_helper_convert_point_meters_2_degrees_array()

Преобразование координат массива точек в метрах (проекции) в координаты градусной меры.

Прототип:

```
#include <gis/gis_math.h>

int gis_helper_convert_point_meters_2_degrees_array( gis_helper_math_ctx_t math_ctx,
↪ double_point_t *out_degrees, uint32_t n_points );
```

Аргументы:

math_ctx

Указатель на математический контекст.

out_degrees

Указатель на исходный массив, в который будет записан результат преобразования, тип `int32_point_t`, `float_point_t`, `double_point_t`

n_points

Количество точек

Библиотека:

gishelper

Описание:

Функция используется для преобразования координат из метров в градусы.

Возвращаемое значение:

EOK

Успешное выполнение

EFAULT

Преобразование завершилось с ошибкой

EINVAL

Некорректный указатель

Примеры использования:

```

gis_helper_math_ctx_t math_ctx;
gis_helper_math_init_ctx( &math_ctx );

gis_map_projection_t projection;
gis_map_projection_init( &projection );
projection.EPSG = 3857;
gis_helper_math_generate_meters_projection( math_ctx, &projection );

double_point_t points[3];
points[0].x = 3000;
points[0].y = 3000;
points[1].x = 4000;
points[1].y = 4000;
points[2].x = 5000;
points[2].y = 5000;

if ( gis_helper_convert_point_meters_2_degrees_array( math_ctx, &points[0], 3 ) != EOK )
↪ {
    printf( "Conversion failed" );
    return 1;
}

if (( fabs( points[0].x - 0.026949 ) < 0.0001 && fabs( points[0].y - 0.026949 ) < 0.0001
↪ ) &&
    ( fabs( points[1].x - 0.035933 ) < 0.0001 && fabs( points[1].y - 0.035933 ) < 0.0001
↪ ) &&
    ( fabs( points[2].x - 0.044916 ) < 0.0001 && fabs( points[2].y - 0.044916 ) < 0.0001
↪ ))
{
    printf( "Correct" );
}
else
{
    printf( "Incorrect degree coordinates: %lf %lf", points[0].x, points[0].y );
    printf( "Incorrect degree coordinates: %lf %lf", points[1].x, points[1].y );
    printf( "Incorrect degree coordinates: %lf %lf", points[2].x, points[2].y );
    return 1;
}

```

Классификация:

ПК ЦКИ для ЗОСРВ «Нейтрино»

Безопасность использования	
Обработчик прерываний	Нет
Обработчик сигналов	Нет
В потоке	Да

Тематические ссылки:

`gis_helper_math_ctx_t, gis_helper_convert_point_degrees_2_meters()`

gis_helper_env_get_config_directory()

Получение пути к директории, содержащей конфигурационные файлы.

Прототип:

```
#include <gis/gishelper.h>

const char *gis_helper_env_get_config_directory();
```

Библиотека:

gishelper

Описание:

Функция возвращает путь к директории, хранящей конфигурационные файлы. По умолчанию используется "\$GIS_ROOT/data/config".

Возвращаемое значение:

!NULL

Путь к директории, хранящей конфигурационные файлы.

Примеры использования:

```
const char *configs = gis_helper_env_get_config_directory();
if ( !configs )
{
    printf( "Config directory is NULL\n" );
    return 1;
}
```

Классификация:

ПК ЦКИ для ЗОСРВ «Нейтрино»

Безопасность использования	
Обработчик прерываний	Нет
Обработчик сигналов	Нет
В потоке	Да

Поддерживается, начиная с ПК ЦКИ редакции 1.1.0

gis_helper_env_get_gis_root_directory()

Получение текущего значения переменной окружения *GIS_ROOT*.

Прототип:

```
#include <gis/gishelper.h>

const char * gis_helper_env_get_gis_root_directory();
```

Библиотека:

gishelper

Описание:

Функция возвращает значение переменной окружения *GIS_ROOT*, задающей путь к картографическому пакету. По умолчанию используется *"/opt/gis"*.

Возвращаемое значение:

!NULL

Значение переменной окружения *GIS_ROOT*.

Примеры использования:

```
const char *gis_root = gis_helper_env_get_gis_root_directory();
const char *getenv_gis_root = std::getenv("GIS_ROOT");

if ( strcmp( gis_root, getenv_gis_root ) != 0 )
{
    printf( "Mismatching envvar GIS_ROOT\n" );
    return 1;
}
```

Классификация:

ПК ЦКИ для ЗОСРВ «Нейтрино»

Безопасность использования	
Обработчик прерываний	Нет
Обработчик сигналов	Нет
В потоке	Да

gis_helper_env_get_maps_cache_directory()

Получение текущего значения переменной окружения `GIS_CORE_MAP_CACHE`.

Прототип:

```
#include <gis/gishelper.h>

const char * gis_helper_env_get_maps_cache_directory();
```

Библиотека:

gishelper

Описание:

Функция возвращает значение переменной окружения `GIS_CORE_MAP_CACHE`. По умолчанию используется `"$GIS_ROOT/data/maps/cache"`.

Возвращаемое значение:

`!NULL`

Значение переменной окружения `GIS_CORE_MAP_CACHE`.

Примеры использования:

```
const char *cache = gis_helper_env_get_maps_cache_directory();
const char *getenv_gis_cache = std::getenv("GIS_CORE_MAP_CACHE");

if ( strcmp( cache, getenv_gis_cache ) != 0 )
{
    printf( "Mismatching envvar GIS_CORE_MAP_CACHE\n" );
    return 1;
}
```

Классификация:

ПК ЦКИ для ЗОСРВ «Нейтрино»

Безопасность использования	
Обработчик прерываний	Нет
Обработчик сигналов	Нет
В потоке	Да

Поддерживается, начиная с ПК ЦКИ редакции 1.1.0

gis_helper_free_string_list()

Освобождение памяти списка строк.

Прототип:

```
#include <gis/gishelper.h>

void gis_helper_free_string_list( char ***str_list );
```

Аргументы:

str_list

Список строк.

Библиотека:

gishelper

Описание:

Функция освобождает память каждой строки типа char* и память указателя char**.

Примеры использования:

```
char str[10] = "0,1,2,3,4";

char **str_list = gis_helper_split_string_by_delimeter( str, ',' );

if ( !str_list )
{
    return 1;
}

gis_helper_free_string_list( &str_list );

if ( str_list != NULL )
{
    return 1;
}

return 0;
```

Классификация:

ПК ЦКИ для ЗОСРВ «Нейтрино»

Безопасность использования	
Обработчик прерываний	Нет
Обработчик сигналов	Да
В потоке	Да

Поддерживается, начиная с ПК ЦКИ редакции 1.4

Тематические ссылки:

`gis_helper_split_string_by_delimeter()`

gis_helper_math_free_ctx()

Освобождение математического контекста.

Прототип:

```
#include <gis/gis_math.h>

void gis_helper_math_free_ctx( gis_helper_math_ctx_t *math_ctx );
```

Аргументы:

math_ctx

Указатель на математический контекст.

Библиотека:

gishelper

Описание:

Функция освобождает ранее инициализированный математический контекст.

Примеры использования:

```
gis_helper_math_ctx_t ctx;

if ( gis_helper_math_init_ctx( &ctx ) != EOK )
{
    printf( "gis_helper_math_init_ctx() failed\n");
    return 1;
}

gis_helper_math_free_ctx( &ctx );
```

Классификация:

ПК ЦКИ для ЗОСРВ «Нейтрино»

Безопасность использования	
Обработчик прерываний	Да
Обработчик сигналов	Нет
В потоке	Да

Тематические ссылки:

`gis_helper_math_ctx_t`, `gis_helper_math_init_ctx()`

gis_helper_math_generate_meters_projection()

Заполнение метрических параметров математического контекста по переданным параметрам проекции.

Прототип:

```
#include <gis/gis_math.h>

int gis_helper_math_generate_meters_projection( gis_helper_math_ctx_t math_ctx,
↪ gis_map_projection_t *parameters );
```

Аргументы:

math_ctx

Указатель на математический контекст.

parameters

Указатель на структуру параметров проекции тип `gis_map_projection_t`

Библиотека:

gishelper

Описание:

Функция заполняет метрические параметры математического контекста по переданным параметрам проекции.

Возвращаемое значение:

EOK

Успешное выполнение

ENOSYS

Некорректные параметры проекции

EINVAL

Некорректный указатель

Примеры использования:

```

gis_helper_math_ctx_t    math_ctx;
gis_map_projection_t     projection;
gis_helper_math_init_ctx( &math_ctx );
gis_map_projection_init( &projection );

if ( gis_helper_math_generate_meters_projection( math_ctx, &projection ) != EOK ) {
    printf( "Failed to initialize metric projection\n" );
    return 1;
}

if ( projection.projection_idx == 5 &&
    projection.ellipsoid_idx == 3 &&
    projection.EPSG == 3395 )
{
    printf( "Correct\n" );
}
else
{
    printf( "Incorrect projection: %d %d %d", projection.projection_idx,
            projection.ellipsoid_idx,
            projection.EPSG );
}

```

Классификация:

ПК ЦКИ для ЗОСРВ «Нейтрино»

Безопасность использования	
Обработчик прерываний	Нет
Обработчик сигналов	Нет
В потоке	Да

Тематические ссылки:

[gis_helper_math_ctx_t](#),

gis_helper_math_get_degrees_projection()

Получение строки, описывающей градусную проекцию, в формате PROJ.4.

Прототип:

```
#include <gis/gis_math.h>

const char * gis_helper_math_get_degrees_projection( gis_helper_math_ctx_t math_ctx );
```

Аргументы:

math_ctx

Указатель на математический контекст.

Библиотека:

gishelper

Описание:

Функция возвращает строку, описывающую градусную проекцию, в формате PROJ.4.

Возвращаемое значение:

!NULL

Строка, описывающая проекцию

NULL

Некорректный указатель

Примеры использования:

```
gis_map_projection_t    projection;
gis_helper_math_ctx_t   math_ctx;

gis_map_projection_init( &projection );
gis_helper_math_init_ctx( &math_ctx );
gis_helper_math_generate_meters_projection( math_ctx, &projection );

if ( strcmp ( gis_helper_math_get_degrees_projection( math_ctx ), "+proj=longlat
↪ +datum=WGS84 +no_defs +type=crs" ) != 0 ) {
    printf( "Current projection: %s\n", gis_helper_math_get_degrees_projection( math_ctx
↪ ) );
```

```
    return 1;  
}
```

Классификация:

ПК ЦКИ для ЗОСРВ «Нейтрино»

Безопасность использования	
Обработчик прерываний	Нет
Обработчик сигналов	Нет
В потоке	Да

Тематические ссылки:

`gis_helper_math_ctx_t, gis_helper_math_generate_meters_projection()`

gis_helper_math_get_meters_projection()

Получение строки, описывающей метрическую проекцию, в формате PROJ.4.

Прототип:

```
#include <gis/gis_math.h>

const char * gis_helper_math_get_meters_projection( gis_helper_math_ctx_t math_ctx );
```

Аргументы:

math_ctx

Указатель на математический контекст.

Библиотека:

gishelper

Описание:

Функция возвращает строку, описывающую метрическую проекцию, в формате PROJ.4.

Возвращаемое значение:

!NULL

Строка, описывающая проекцию

NULL

Некорректный указатель

Примеры использования:

```
gis_map_projection_t    projection;
gis_helper_math_ctx_t   math_ctx;

gis_map_projection_init( &projection );
gis_helper_math_init_ctx( &math_ctx );
gis_helper_math_generate_meters_projection( math_ctx, &projection );

if ( strcmp ( gis_helper_math_get_meters_projection( math_ctx ), "+proj=merc +lon_0=0
↪ +k=1 +x_0=0 +y_0=0 +datum=WGS84 +units=m +no_defs +type=crs" ) != 0 ) {
    printf( "Current projection: %s", gis_helper_math_get_meters_projection( math_ctx )
↪ );
```

```
    return 1;  
}
```

Классификация:

ПК ЦКИ для ЗОСРВ «Нейтрино»

Безопасность использования	
Обработчик прерываний	Нет
Обработчик сигналов	Нет
В потоке	Да

Тематические ссылки:

`gis_helper_math_ctx_t, gis_helper_math_generate_meters_projection()`

gis_helper_math_init_ctx()

Инициализация математического контекста.

Прототип:

```
#include <gis/gis_math.h>

int gis_helper_math_init_ctx( gis_helper_math_ctx_t *math_ctx );
```

Аргументы:

math_ctx

Указатель на математический контекст.

Библиотека:

gishelper

Описание:

Функция инициализирует математический контекст, необходимый для преобразований координат и работы с ними.

Возвращаемое значение:

EOK

Успешная инициализация

ENOMEM

Недостаточно памяти

EINVAL

Некорректный указатель

Примеры использования:

```
gis_helper_math_ctx_t ctx;

if ( gis_helper_math_init_ctx( &ctx ) != EOK )
{
    printf( "gis_helper_math_init_ctx() failed\n");
    return 1;
}

gis_helper_math_free_ctx( &ctx );
```

Классификация:

ПК ЦКИ для ЗОСРВ «Нейтрино»

Безопасность использования	
Обработчик прерываний	Нет
Обработчик сигналов	Нет
В потоке	Да

Тематические ссылки:

`gis_helper_math_ctx_t`, `gis_helper_math_free_ctx()`

gis_helper_split_string_by_delimeter()

Разбить строку по разделителю.

Прототип:

```
#include <gis/gishelper.h>

char **gis_helper_split_string_by_delimeter( char* source_string, const char delimiter );
```

Аргументы:

source_string
Список строк.

delimiter
Разделитель.

Библиотека:

gishelper

Описание:

Функция разбивает строку на подстроки по разделителю `a_delim`, и возвращает список строк. Функция не изменяет исходную строку.

Возвращаемое значение:

nullptr
Функция завершилась с ошибкой.



Функция выделяет память для списка строк. Для освобождения памяти исп. [gis_helper_free_string_list\(\)](#).

Примеры использования:

```

char str[10] = "0,1,2,3,4";

char **str_list = gis_helper_split_string_by_delimeter( str, ',' );

if ( !str_list )
{
    return 1;
}

gis_helper_free_string_list( &str_list );

return 0;

```

Классификация:

ПК ЦКИ для ЗОСРВ «Нейтрино»

Безопасность использования	
Обработчик прерываний	Нет
Обработчик сигналов	Да
В потоке	Да

Поддерживается, начиная с ПК ЦКИ редакции 1.4

Тематические ссылки:

[gis_helper_free_string_list\(\)](#)

gis_object_primitive_type_get_full_name()

Получение строки с названием типа примитива объекта карты.

Прототип:

```
#include <gis/gis_types.h>

static inline const char * gis_object_primitive_type_get_full_name( uint32_t type );
```

Аргументы:

type

Тип примитива объекта карты.

Библиотека:

gishelper

Описание:

Функция возвращает статическую строку, описывающую название типа примитива объекта карты.

Возвращаемое значение:

!"Unknown"

Строка с названием типа примитива объекта карты

Unknown

Передан некорректный тип примитива

Примеры использования:

```
gis_object_primitive_type_t type = GIS_OBJECT_PRIMITIVE_POLYGON;

if ( strcmp( gis_object_primitive_type_get_full_name( type ), "Polygon" ) != 0 )
{
    printf( "gis_object_primitive_type_get_full_name() failed\n" );
    return 1;
}
```

Классификация:

Безопасность использования	
Обработчик прерываний	Да
Обработчик сигналов	Да
В потоке	Да

Тематические ссылки:

[gis_object_primitive_type_t](#)

gis_borders_t

Границы картографической информации.

Формат:

```
#include <gis/gis_types.h>

typedef union {
    struct gis_borders_degree {
        double    west,
                 south,
                 east,
                 north;
    }             degrees;
} gis_borders_t;
```

Описание:

Данный тип данных обеспечивает определение географических границ ЦКИ библиотекой gishelper. Поле degrees позволяет задать географические границы в градусах:

Аргументы:

west

Левая граница прямоугольной области.

south

Нижняя граница прямоугольной области.

east

Правая граница прямоугольной области.

north

Верхняя граница прямоугольной области.

Классификация:

ПК ЦКИ для ЗОСРВ «Нейтрино»

gis_data_raw_context_t

Контекст низкоуровневого доступа к данным.

Формат:

```
#include <gis/gis_raw.h>

typedef void * gis_data_raw_context_t;
```

Описание:

Данный контекст обеспечивает выполнение низкоуровневых операций доступа к ЦКИ средствами библиотеки gishelper.

Классификация:

ПК ЦКИ для ЗОСРВ «Нейтрино»

gis_helper_math_ctx_t

Контекст математических операций.

Формат:

```
#include <gis/gis_math.h>

typedef void * gis_helper_math_ctx_t;
```

Описание:

Данный контекст обеспечивает выполнение математических операций библиотекой gishelper.

Классификация:

ПК ЦКИ для ЗОСРВ «Нейтрино»

gis_object_primitive_type_t

Типы примитивов объекта карты.

Формат:

```
#include <gis/gis_types.h>

typedef enum {
    GIS_OBJECT_PRIMITIVE_POLYGON      = 0x0,
    GIS_OBJECT_PRIMITIVE_LINE         = 0x1,
    GIS_OBJECT_PRIMITIVE_POINT        = 0x2,
    GIS_OBJECT_PRIMITIVE_TEXT         = 0x3,
    GIS_OBJECT_PRIMITIVE_VECTOR       = 0x4,
    GIS_OBJECT_PRIMITIVE_TEXT_TEMPLATE = 0x5,
    GIS_OBJECT_PRIMITIVE_UNKNOWN      = GIS_OBJECT_PRIMITIVE_TEXT_TEMPLATE + 1,
    GIS_OBJECT_PRIMITIVE_COUNT        = GIS_OBJECT_PRIMITIVE_UNKNOWN + 1
} gis_object_primitive_type_t;
```

Описание:

Перечисление gis_object_primitive_type_t включают следующие значения:

- GIS_OBJECT_PRIMITIVE_POINT — точка
- GIS_OBJECT_PRIMITIVE_LINE — линия
- GIS_OBJECT_PRIMITIVE_POLYGON — полигон
- GIS_OBJECT_PRIMITIVE_TEXT — текст
- GIS_OBJECT_PRIMITIVE_VECTOR — вектор
- GIS_OBJECT_PRIMITIVE_TEXT_TEMPLATE — шаблон текста
- GIS_OBJECT_PRIMITIVE_UNKNOWN - для внутреннего использования
- GIS_OBJECT_PRIMITIVE_COUNT - для внутреннего использования

Классификация:

ПК ЦКИ для ЗОСРВ «Нейтрино»

int32_point_t, float_point_t, double_point_t

Представление примитива 'точка'.

Формат:

```
#include <gis/gis_types.h>

typedef struct {
    int32_t    x;
    int32_t    y;
} int32_point_t;

typedef struct {
    float      x;
    float      y;
} float_point_t;

typedef struct {
    double     x;
    double     y;
} double_point_t;
```

Описание:

Типы int32_point_t, float_point_t, double_point_t описывают геометрический примитив 'точка'.

Классификация:

ПК ЦКИ для ЗОСРВ «Нейтрино»

Тематические ссылки:

[gis_helper_convert_point_meters_2_degrees\(\)](#), [gis_helper_convert_point_degrees_2_meters\(\)](#)

object_point_t

Представление 'точки' объекта.

Формат:

```
#include <gis/gis_types.h>

typedef struct
{
    double_point_t degrees;
    double_point_t meters;
} object_point_t;
```

Описание:

Тип описывает хранимые координаты точки объекта.

Аргументы:

degrees

Координаты в градусах.

meters

Метрические координаты конкретной проекции.

Классификация:

ПК ЦКИ для ЗОСРВ «Нейтрино»

Тематические ссылки:

[int32_point_t](#), [float_point_t](#), [double_point_t](#)

CLIP_VALUE

Контроль нахождения в диапазоне между двумя числами.

Формат:

```
#include <gis/gis_math.h>

#define CLIP_VALUE(x, lower, upper)      (min( (upper), max( (x), (lower) ) ))
```

Аргументы:

<i>x</i>	Искомое число для проверки.
<i>lower</i>	Нижняя граница диапазона.
<i>upper</i>	Верхняя граница диапазона.

Библиотека:

gishelper

Описание:

Макрос определяет, входит ли переданное число в переданный диапазон чисел. Если входит, то оно же и возвращается обратно. Если не входит, то обратно возвращается ближайшая граница диапазона.

Возвращаемое значение:

<i>value</i>	Искомое число либо ближайшая к нему граница диапазона.
--------------	--

Примеры использования:

```
double north;  
  
north = 91;  
north = CLIP_VALUE( north, -90, 90 ); // 90  
  
north = -91;  
north = CLIP_VALUE( north, -90, 90 ); // -90  
  
north = 89;  
north = CLIP_VALUE( north, -90, 90 ); // 89  
  
north = -89;  
north = CLIP_VALUE( north, -90, 90 ); // -89
```

Классификация:

ПК ЦКИ для ЗОСРВ «Нейтрино»

DEGREES_TO_RADS

Конвертация градусных углов в радианы.

Формат:

```
#include <gis/gis_math.h>

#define DEGREES_TO_RADS( grads )      ((grads) * 1.0 * M_PI / 180.0)
```

Аргументы:

grads

Угол в градусах.

Библиотека:

gishelper

Описание:

Макрос конвертирует переданное числовое значение из градусов в радианы.

Аргументы:

rads

Угол в радианах.

Примеры использования:

```
double grads, rads;
grads = 83.2;
rads = DEGREES_TO_RADS( grads );
```

Классификация:

ПК ЦКИ для ЗОСРВ «Нейтрино»

EQUAL_POINTS_XY

Проверка двух координат на равенство.

Формат:

```
#include <gis/gis_math.h>

#define EQUAL_POINTS_XY( p1, p2 )      ((p1).x == (p2).x && (p1).y == (p2).y)
```

Аргументы:

p1

Первая искомая координата. См. `int32_point_t`, `float_point_t`, `double_point_t`.

p2

Вторая искомая координата. См. `int32_point_t`, `float_point_t`, `double_point_t`.

Библиотека:

gishelper

Описание:

Макрос позволяет проверить, являются ли две координаты одинаковыми.

Возвращаемое значение:

0

Координаты неодинаковы.

1

Координаты одинаковы.

Примеры использования:

```
double_point_t pt1 = {2.0, 3.0};
double_point_t pt2 = {2.0, 3.0};

if ( EQUAL_POINTS_XY( pt1, pt2 ) )
{
    printf( "Points are equal!\n" );
}
```

Классификация:

ПК ЦКИ для ЗОСРВ «Нейтрино»

EQUAL_POINTS_XY_PTRS

Проверка двух координат на равенство с использованием указателей.

Формат:

```
#include <gis/gis_math.h>

#define EQUAL_POINTS_XY_PTRS( pptr1, pptr2 )    ((pptr1)->x == (pptr2)->x && (pptr1)->y == \
↪ (pptr2)->y)
```

Аргументы:

pptr1

Указатель на первую искомую координату. См. [int32_point_t](#), [float_point_t](#), [double_point_t](#).

pptr2

Указатель на вторую искомую координату. См. [int32_point_t](#), [float_point_t](#), [double_point_t](#).

Библиотека:

gishelper

Описание:

Макрос позволяет проверить, являются ли две координаты одинаковыми, используя указатели на обе координаты.

Возвращаемое значение:

0

Координаты неодинаковы.

1

Координаты одинаковы.

Примеры использования:

```
double_point_t pt1 = {2.0, 3.0};
double_point_t pt2 = {2.0, 3.0};

if ( EQUAL_POINTS_XY_PTRS( &pt1, &pt2 ) )
{
    printf( "Points are equal!\n" );
}
```

Классификация:

ПК ЦКИ для ЗОСРВ «Нейтрино»

GETSIGN

Получение знака числа.

Формат:

```
#include <gis/gis_math.h>

#define GETSIGN( x )      ((x) > 0 ? 1 : (x) < 0 ? -1 : 0)
```

Аргументы:

x
Искомое число.

Библиотека:

gishelper

Описание:

Макрос позволяет получить знак числа.

Возвращаемое значение:

-1
Искомое число отрицательно.

0
Искомое число - нуль.

1
Искомое число положительно.

Примеры использования:

```
double x, sign;

x = 2;
sign = GETSIGN( x ); // 1

x = -2;
sign = GETSIGN( x ); // -1

x = 0;
sign = GETSIGN( x ); // 0
```

Классификация:

ПК ЦКИ для ЗОСРВ «Нейтрино»

Поддерживается, начиная с ПК ЦКИ редакции 1.4



До версии 1.4 - SIGN

RADS_TO_DEGREES

Конвертация радиан в градусные углы.

Формат:

```
#include <gis/gis_math.h>

#define RADS_TO_DEGREES( rads )      ((rads) * 1.0 * 180.0 / M_PI)
```

Аргументы:

rads

Угол в радианах.

Библиотека:

gishelper

Описание:

Макрос конвертирует переданное числовое значение из радиан в градусы.

Возвращаемое значение:

grads

Угол в градусах.

Примеры использования:

```
double grads, rads;
rads = 1.03;
grads = RADS_TO_DEGREES( rads );
```

Классификация:

ПК ЦКИ для ЗОСРВ «Нейтрино»

gis_common_driver_check_folders

Проверить наличие папок с картами и стилями.

Прототип:

```
#include <gis/gis_driver.h>

int gis_common_driver_check_folders( gis_driver_ctx_t *driver_ctx );
```

Аргументы:

driver_ctx

Контекст драйвера.

Библиотека:

formatctl

Описание:

Функция проверяет наличие `origin_maps_dirname` и `mapstyle_dirname`, и создает эти папки, если их нет. Наличие `mapstyle_dirname` не проверяется, если значение равно `NULL`.

Примеры использования:

```
gis_driver_ctx_t driver_ctx;

if ( gis_common_driver_get_mandatory_driver_param_from_cfg(
↪ &driver_ctx.origin_maps_dirname,
                                                                    "jpeg2000-local",
                                                                    "map_folder",
                                                                    "[JPEG2000_local_driver] " ) !=
↪ EOK )
{
    printf( "gis_common_driver_check_folders:
↪ gis_common_driver_get_mandatory_driver_param_from_cfg() failed!\n");
    return 1;
}

if ( gis_common_driver_check_folders( &driver_ctx ) != EOK )
{
    free( (char *)driver_ctx.origin_maps_dirname );
    printf( "gis_common_driver_check_folders failed!\n");
    return 1;
}
```

```
}  
  
free( (char *)driver_ctx.origin_maps_dirname );
```

Классификация:

ПК ЦКИ для ЗОСРВ «Нейтрино»

Безопасность использования	
Обработчик прерываний	Нет
Обработчик сигналов	Нет
В потоке	Да

Поддерживается, начиная с ПК ЦКИ редакции 1.3

Тематические ссылки:

gis_driver_ctx_t

gis_common_driver_fill_extensions()

Заполнить список расширений карт.

Прототип:

```
#include <gis/gis_driver.h>

int gis_common_driver_fill_extensions( gis_common_driver_extension_list_t *extensions_list,
↪ char *extensions_cfg_string );
```

Аргументы:

extensions_list

Список расширений (`gis_common_driver_extension_list_t`).

extensions_cfg_string

Строка с расширениями (разделитель - пробел).

Библиотека:

formatctl

Описание:

Функция динамически выделяет память для массива и его элементов, заполняет его подстроками, полученными из `extensions_cfg_string`.

Примеры использования:

```
gis_common_driver_extension_list_t alternative_extensions;

const char *extensions_cfg_string = "jpg j2k jpf jpm jpg2 j2c jpc jpx mj2";

if ( gis_common_driver_fill_extensions( &alternative_extensions, (char
↪ *)extensions_cfg_string ) != EOK )
{
    printf( "gis_common_driver_fill_extensions() failed!\n");
    return 1;
}

for ( int i = 0; i < alternative_extensions.count; i++ )
{
    free( (char *)alternative_extensions.extensions[i] );
}
```

```
free( (char **)alternative_extensions.extensions );
```

Классификация:

ПК ЦКИ для ЗОСРВ «Нейтрино»

Безопасность использования	
Обработчик прерываний	Нет
Обработчик сигналов	Нет
В потоке	Да

Поддерживается, начиная с ПК ЦКИ редакции 1.3

Тематические ссылки:

`gis_common_driver_extension_list_t`

gis_common_driver_get_basename_without_extension

Получить базовое имя карты без расширения.

Прототип:

```
#include <gis/gis_driver.h>

int gis_common_driver_get_basename_without_extension( const char *map_filepath, char
↳ (*result)[GIS_MAX_PATH_LENGTH] );
```

Аргументы:

map_filepath

Путь к карте (map_info->src_fname).

result

Базовое имя карты без расширения.

Библиотека:

formatctl

Описание:

Функция отбрасывает расширение и возвращает базовое имя карты.

Примеры использования:

```
char orig_basename[GIS_MAX_PATH_LENGTH];

if ( gis_common_driver_get_basename_without_extension( "filepath.ext", &orig_basename ) !=
↳ EOK )
{
    printf( "gis_common_driver_get_basename_without_extension failed!\n");
    return 1;
}
```

Классификация:

ПК ЦКИ для ЗОСРВ «Нейтрино»

Безопасность использования	
Обработчик прерываний	Нет
Обработчик сигналов	Нет
В потоке	Да

Поддерживается, начиная с ПК ЦКИ редакции 1.3

Тематические ссылки:

[gis_core_map_information_t](#)

gis_common_driver_get_mandatory_driver_param_from_cfg()

Получить из файла конфигурации ядра один из основных (обязательных) параметров драйвера.

Прототип:

```
#include <gis/gis_driver.h>

int gis_common_driver_get_mandatory_driver_param_from_cfg( const char **param_to_init,
                                                           const char *section_key,
                                                           const char *param_key,
                                                           const char *prefix );
```

Аргументы:

param_to_init

Элемент структуры `gis_driver_ctx_t`.

section_key

Название раздела файла конфигурации ядра.

param_key

Название атрибута в файле конфигурации ядра, значение которого требуется получить.

prefix

Префикс драйвера, используемый в отладочном выводе.

Библиотека:

formatctl

Описание:

Функция динамически выделяет память для строки и возвращает прочитанное значение. Искомый параметр, значение которого возвращается, обязан быть в файле конфигурации ядра.

Примеры использования:

```
gis_driver_ctx_t driver_ctx;

if ( gis_common_driver_get_mandatory_driver_param_from_cfg( &driver_ctx.origin_extension,
                                                           "jpeg2000-local",
                                                           "fmt_extension",
                                                           "[JPEG2000_local_driver] " ) !=
↳ EOK )
{
```

```

    printf( "gis_common_driver_get_mandatory_driver_param_from_cfg() failed!\n");
    return 1;
}

free( (char *)driver_ctx.origin_extension );

```

Классификация:

ПК ЦКИ для ЗОСРВ «Нейтрино»

Безопасность использования	
Обработчик прерываний	Нет
Обработчик сигналов	Нет
В потоке	Да

Поддерживается, начиная с ПК ЦКИ редакции 1.3

Тематические ссылки:

[gis_driver_ctx_t](#)

gis_common_driver_recover_gcm_database()

Восстановить базу данных GCM.

Прототип:

```
#include <gis/gis_driver.h>

int gis_common_driver_recover_gcm_database( gis_driver_ctx_t *driver_ctx );
```

Аргументы:

driver_ctx

Контекст драйвера.

Библиотека:

formatctl

Описание:

Функция пытается добавить в базу данных GCM записи из папки /gcm/. Если валидация заголовка gcm провалилась, функция пытается переконвертировать карту. Если попытка конвертации провалилась, файл GCM удаляется.

Примеры использования:

```
gis_driver_ctx_t driver_ctx;
driver_ctx.id = GIS_CORE_DRIVERS_SXF_LOCAL;
driver_ctx.prefix = "[SXF_local_driver] ";

char *suffix = calloc( strlen( "sxf" ) + strlen( ".local."GIS_GCM_FILE_EXTENSION ) + 1,
↳ sizeof( char ) );

if ( !suffix )
{
    gis_helper_debug_write( GIS_DEBUG_LEVEL_ERROR, driver_ctx.prefix,
        "Memory allocation error [%s()]", __FUNCTION__ );
    return ENOMEM;
}

strcpy( suffix, "sxf" );
driver_ctx.driver_gcm_suffix = strcat( suffix, ".local."GIS_GCM_FILE_EXTENSION );

if ( gis_common_driver_recover_gcm_database( driver_ctx ) != EOK )
```

```

{
    gis_helper_debug_write_lvl( GIS_DEBUG_LEVEL_ERROR, driver_ctx.prefix,
                                "Failed to recover GCM database [%s()]", __FUNCTION__ );
    return EFAULT;
}

free( (char*)driver_ctx.driver_gcm_suffix );

```

Классификация:

ПК ЦКИ для ЗОСРВ «Нейтрино»

Безопасность использования	
Обработчик прерываний	Нет
Обработчик сигналов	Нет
В потоке	Да

Поддерживается, начиная с ПК ЦКИ редакции 1.3

Тематические ссылки:

[gis_driver_ctx_t](#)

gis_common_driver_reset_gcm_database()

Очистить базу данных GCM.

Прототип:

```
#include <gis/gis_driver.h>

int gis_common_driver_reset_gcm_database( gis_core_driver_id_t id );
```

Аргументы:

id
Идентификатор драйвера.

Библиотека:

formatctl

Описание:

Функция используется для удаления имеющихся GCM записей одного исходного формата (hard reset).

Примеры использования:

```
gis_driver_ctx_t driver_ctx;
driver_ctx.id = GIS_CORE_DRIVERS_SXF_LOCAL;
bool sync_initial_source_is_hard = true;

if ( sync_initial_source_is_hard )
{
    gis_common_driver_reset_gcm_database( driver_ctx.id );
}
```

Классификация:

ПК ЦКИ для ЗОСРВ «Нейтрино»

Безопасность использования	
Обработчик прерываний	Нет
Обработчик сигналов	Нет
В потоке	Да

Поддерживается, начиная с ПК ЦКИ редакции 1.3

Тематические ссылки:

gis_core_driver_id_t

gis_common_driver_sync_cache()

Синхронизировать кэш карт.

Прототип:

```
#include <gis/gis_driver.h>

int gis_common_driver_sync_cache( gis_driver_ctx_t *driver_ctx , int remove_flag );
```

Аргументы:

driver_ctx

Контекст драйвера.

remove_flag

Если GCM существует, а src отсутствует и установлен <remove_flag>, то GCM удаляется.

Библиотека:

formatctl

Описание:

Функция синхронизирует список GCM карт.

Примеры использования:

```
gis_driver_ctx_t driver_ctx; // must be initialized
int res = gis_common_driver_sync_cache( &driver_ctx, 0 );
if ( res != EOK )
{
    gis_helper_debug_write_lvl( GIS_DEBUG_LEVEL_ERROR, "DRIVER_PREFIX",
                                "Failed to synchronize cache (err %d) [%s()]",
                                res, __FUNCTION__ );
    return res;
}
```

Классификация:

ПК ЦКИ для ЗОСРВ «Нейтрино»

Безопасность использования	
<i>Обработчик прерываний</i>	Нет
<i>Обработчик сигналов</i>	Нет
<i>В потоке</i>	Да

Поддерживается, начиная с ПК ЦКИ редакции 1.3

Тематические ссылки:

gis_driver_ctx_t

gis_common_driver_update_cache()

Переконвертировать карту в GCM.

Прототип:

```
#include <gis/gis_driver.h>

int gis_common_driver_update_cache( gis_driver_ctx_t *driver_ctx, int32_t map_id );
```

Аргументы:

driver_ctx

Контекст драйвера.

map_id

Идентификатор карты. Если *map_id* <0 -> функция работает со всеми картами для заданного драйвера.

Библиотека:

formatctl

Описание:

Функция удаляет карту из базы данных GCM и файловой системы, затем конвертирует карту в GCM.

Примеры использования:

```
gis_driver_ctx_t driver_ctx; // must be initialized
int res = gis_common_driver_update_cache( &driver_ctx, 0 );
if ( res != EOK )
{
    gis_helper_debug_write_lvl( GIS_DEBUG_LEVEL_ERROR, "DRIVER_PREFIX",
                               "Failed to update cache (err %d) [%s()]",
                               res, __FUNCTION__ );
    return res;
}
```

Классификация:

ПК ЦКИ для ЗОСРВ «Нейтрино»

Безопасность использования	
Обработчик прерываний	Нет
Обработчик сигналов	Нет
В потоке	Да

Поддерживается, начиная с ПК ЦКИ редакции 1.3

Тематические ссылки:

[gis_driver_ctx_t](#)

gis_common_driver_update_clipped_map_info()

Обновить информацию о картах GCM с одноимённой исходной картой.

Прототип:

```
#include <gis/gis_driver.h>

int gis_common_driver_update_clipped_map_info( gis_driver_ctx_t *driver_ctx,
↪ gis_core_map_information_t *map_info );
```

Аргументы:

driver_ctx

Контекст драйвера.

map_info

Структура `gis_core_map_information_t`.

Библиотека:

formatctl

Описание:

Функция обновляет поля `gcm_clip_*` в структуре `map_info`.

Примеры использования:

```
gis_driver_ctx_t driver_ctx;
driver_ctx.id = GIS_CORE_DRIVERS_SXF_LOCAL;
driver_ctx.prefix = "[SXF_local_driver] ";

gis_core_map_information_t *map_info; // must be initialized

char *suffix = calloc( strlen( "sxf" ) + strlen( ".local."GIS_GCM_FILE_EXTENSION ) + 1,
↪ sizeof( char ) );

if ( !suffix )
{
    gis_helper_debug_write( GIS_DEBUG_LEVEL_ERROR, DRIVER_PREFIX,
        "Memory allocation error [%s()]", __FUNCTION__ );
    return ENOMEM;
}
```

```

strcpy( suffix, "sxf" );
driver_ctx.driver_gcm_suffix = strcat( suffix, ".local."GIS_GCM_FILE_EXTENSION );

if ( gis_common_driver_update_clipped_map_info( &driver_ctx, map_info ) != EOK )
{
    gis_helper_debug_write_lvl( GIS_DEBUG_LEVEL_ERROR, driver_ctx.prefix,
                                "Update clipped mapinfo failed [%s()]", __FUNCTION__ );
    return EFAULT;
}

free( (char*)driver_ctx.driver_gcm_suffix );

```

Классификация:

ПК ЦКИ для ЗОСРВ «Нейтрино»

Безопасность использования	
Обработчик прерываний	Нет
Обработчик сигналов	Нет
В потоке	Да

Поддерживается, начиная с ПК ЦКИ редакции 1.3

Тематические ссылки:

[gis_driver_ctx_t](#), [gis_core_map_information_t](#)

gis_common_driver_extension_list_t

Список возможных расширений файлов карт.

Формат:

```
#include <gis/gis_driver.h>

typedef struct
{
    int count;
    const char **extensions;
} gis_common_driver_extension_list_t;
```

Описание:

Структура определяет список возможных расширений файлов карт.

Аргументы:

count

Количество элементов массива.

extensions

Массив, содержащий строки с расширениями.

Классификация:

ПК ЦКИ для ЗОСРВ «Нейтрино»

Поддерживается, начиная с ПК ЦКИ редакции 1.3

Тематические ссылки:

[gis_driver_ctx_t](#)

gis_driver_ctx_t

Контекст драйвера формата карт.

Формат:

```
#include <gis/gis_driver.h>

typedef struct
{
    gis_core_driver_id_t      id;
    bool                     initialized;
    gis_core_connection_state_t state;
    const char               *acronym;
    const char               *prefix;
    const char               *origin_maps_dirname;
    const char               *mapstyle_dirname;
    const char               *origin_extension;
    gis_common_driver_extension_list_t *alt_origin_extensions;
    const char               *driver_gcm_suffix;

    const char               *converter_path;

    gis_common_driver_extension_list_t *secondary_extensions;

    int (*convert_map)(gis_core_map_information_t *);
} gis_driver_ctx_t;
```

Описание:

Структура определяет контекст драйвера.

Аргументы:

id

Уникальный идентификатор драйвера.

initialized

Флаг, устанавливаемый после окончания инициализации драйвера.

state

Текущее состояние подключения к картографическому сервису (ядру).

acronym

Акроним драйвера.

prefix

Префикс драйвера, используемый в отладочном выводе.

origin_maps_dirname

Название директории с картами; должна находиться внутри директории `$GIS_CORE_MAP_CACHE`.

mapstyle_dirname

Название директории со стилями карты; должна находиться внутри `origin_maps_dirname`.

origin_extension

Расширение карт, используемое по умолчанию.

alt_origin_extensions

Альтернативные расширения карт.

driver_gcm_suffix

Оригинальное расширение карты + `".local."` + `GIS_GCM_FILE_EXTENSION`.

converter_path

Путь до конвертера карт относительно корня проекта `$GIS_ROOT`.

secondary_extensions

Расширения, используемые в случае, если карта состоит из нескольких файлов.

convert_map

Функция вызова конвертера карты во внутренний формат GCM, принимающая в качестве аргумента (`gis_core_map_information_t`).

Классификация:

ПК ЦКИ для ЗОСРВ «Нейтрино»

Поддерживается, начиная с ПК ЦКИ редакции 1.3

Тематические ссылки:

`gis_core_driver_id_t`, `gis_core_connection_state_t`, `gis_common_driver_extension_list_t`

gis_driver_interface_t

Интерфейс драйвера формата исходной карты.

Формат:

```
#include <gis/gis_driver.h>

typedef struct
{
    int (*connect)( void );
    int (*get_acronym)( char *acronym );
    gis_core_connection_state_t (*get_connection_state)( void );
    int (*get_info)( gis_core_driver_info_t *info );
    int (*reset_cache)( void );
    int (*synchronize_cache)( void );
    int (*update_cache)( int32_t map_id );
} gis_driver_interface_t;
```

Описание:

Структура содержит указатели на функции, по умолчанию требующие реализации в драйвере.

Аргументы:

connect

Инициировать подключение к ядру.

get_acronym

Получить акроним драйвера.

get_connection_state

Получить текущее состояние подключения к картографическому сервису в формате [gis_core_connection_state](#)

get_info

Получить информацию в формате [gis_core_driver_info_t](#).

reset_cache

Удалить все исходные данные и снова кэшировать.

synchronize_cache

Добавить отсутствующие данные в GCM и убрать отсутствующие данные из источников.

update_cache

Переконвертировать карту в GCM (map_id <0 -> Конвертировать все).

Классификация:

ПК ЦКИ для ЗОСРВ «Нейтрино»

Поддерживается, начиная с ПК ЦКИ редакции 1.3

GIS_CACHE_SUBDIR_GCM

Маркер названия директории, хранящей карты формата GCM.

Формат:

```
#define GIS_CACHE_SUBDIR_GCM    "gcm"
```

Библиотека:

formatctl

Описание:

Используется в драйверах для указания пути до карты GCM.

Примеры использования:

```
char gcm_filepath[GIS_MAX_PATH_LENGTH];  
const char *directory = gis_helper_env_get_maps_cache_directory();  
sprintf( gcm_filepath, "%s/%s/%s", directory, GIS_CACHE_SUBDIR_GCM,  
↪ "map_name.origin_extension.local."GIS_GCM_FILE_EXTENSION );
```

Классификация:

ПК ЦКИ для ЗОСРВ «Нейтрино»

Поддерживается, начиная с ПК ЦКИ редакции 1.3

GIS_GCM_FILE_EXTENSION

Маркер расширения карт формата GCM

Формат:

```
#define GIS_GCM_FILE_EXTENSION "gcm"
```

Библиотека:

formatctl

Описание:

Используется в драйверах для инициализации поля driver_gcm_suffix в gis_driver_ctx_t.

Примеры использования:

```
gis_driver_ctx_t driver_ctx;

char *suffix = calloc( strlen( "origin_extension" ) + strlen(
↪ ".local."GIS_GCM_FILE_EXTENSION ) + 1, sizeof( char ) );

if ( !suffix )
{
    gis_helper_debug_write( GIS_DEBUG_LEVEL_ERROR, DRIVER_PREFIX,
        "Memory allocation error [%s()]", __FUNCTION__ );
    return ENOMEM;
}

strcpy( suffix, "origin_extension" );

driver_ctx.driver_gcm_suffix = strcat( suffix, ".local."GIS_GCM_FILE_EXTENSION );
free( (char*)driver_ctx.driver_gcm_suffix );
```

Классификация:

ПК ЦКИ для ЗОСРВ «Нейтрино»

Поддерживается, начиная с ПК ЦКИ редакции 1.3

gis_core_config_get_driver_value()

Получить из файла конфигурации ядра параметр драйвера.

Прототип:

```
#include <gis/gis_core_cfg_parser.h>

int gis_core_config_get_driver_value( const char    *driver_name,
                                     const char    *key
                                     char  (*out_val)[GIS_MAX_PATH_LENGTH] );
```

Аргументы:

driver_name

Название раздела файла конфигурации (name = driver_name).

key

Название атрибута в файле конфигурации, значение которого требуется получить.

out_val

Искомое значение.

Библиотека:

gishelper

Описание:

Функция возвращает значение, прочитанное из файла конфигурации.



Функция не выделяет память для out_val.

Примеры использования:

```

char opt_val[GIS_MAX_PATH_LENGTH];

if ( gis_core_config_get_driver_value( DRIVER_S57_CFG_SECTION,
                                      DRIVER_CFG_CONVERTER_PATH,
                                      &opt_val ) != EOK )
{
    printf( "gis_core_config_get_driver_value() failed!\n");
    return 1;
}

if ( strcmp( opt_val, "sbin/gis-s57-processor", 22 ) != 0 )
{
    printf( "gis_core_config_get_driver_value() found unexpected value!\n");
    return 1;
}

```

Классификация:

ПК ЦКИ для ЗОСРВ «Нейтрино»

Безопасность использования	
Обработчик прерываний	Нет
Обработчик сигналов	Нет
В потоке	Да

Поддерживается, начиная с ПК ЦКИ редакции 1.3

gis_core_config_get_value()

Получить параметр из файла конфигурации ядра.

Прототип:

```
#include <gis/gis_core_cfg_parser.h>

int gis_core_config_get_value( const char *key,
                              char (*out_val)[GIS_MAX_PATH_LENGTH] );
```

Аргументы:

key

Название атрибута в файле конфигурации, значение которого требуется получить.

out_val

Искомое значение.

Библиотека:

gishelper

Описание:

Функция возвращает значение, прочитанное из файла конфигурации.



Функция не выделяет память для `out_val`.

Примеры использования:

```
char config_buf[GIS_MAX_PATH_LENGTH];

if ( gis_core_config_get_value( "name", &config_buf ) != EOK )
{
    printf( "gis_core_config_get_driver_value() failed!\n" );
    return 1;
}
```

Классификация:

Безопасность использования	
<i>Обработчик прерываний</i>	Нет
<i>Обработчик сигналов</i>	Нет
<i>В потоке</i>	Да

Поддерживается, начиная с ПК ЦКИ редакции 1.3

gis_core_config_set_driver_value()

Обновить значение параметра драйвера в файле конфигурации ядра.

Прототип:

```
#include <gis/gis_core_cfg_parser.h>

int gis_core_config_set_driver_value( const char *driver_name,
                                     const char *key,
                                     const char *out_val );
```

Аргументы:

driver_name

Название раздела файла конфигурации (name = driver_name).

key

Искомый параметр.

out_val

Новое значение.

Библиотека:

gishelper

Описание:

Функция создаст новый файл конфигурации с обновленным значением, удалив старый файл.

Примеры использования:

```
if ( gis_core_config_set_driver_value( DRIVER_S57_CFG_SECTION,
                                       DRIVER_CFG_CONVERTER_PATH,
                                       "new path" ) != EOK )
{
    printf( "gis_core_config_set_driver_value() failed!\n");
    return 1;
}

char opt_val[GIS_MAX_PATH_LENGTH];

if ( gis_core_config_get_driver_value( DRIVER_S57_CFG_SECTION,
                                       DRIVER_CFG_CONVERTER_PATH,
                                       &opt_val ) != EOK )
```

```

{
    printf( "gis_core_config_get_driver_value() failed!\n");
    return 1;
}

if ( strncmp( opt_val, "new path", 8 ) != 0 )
{
    printf( "gis_core_config_get_driver_value() found unexpected value!\n");
    return 1;
}

if ( gis_core_config_set_driver_value( DRIVER_S57_CFG_SECTION,
                                      DRIVER_CFG_CONVERTER_PATH,
                                      "sbin/gis-s57-processor" ) != EOK )
{
    printf( "gis_core_config_set_driver_value() failed!\n");
    return 1;
}

```

Классификация:

ПК ЦКИ для ЗОСРВ «Нейтрино»

Безопасность использования	
Обработчик прерываний	Нет
Обработчик сигналов	Нет
В потоке	Да

Поддерживается, начиная с ПК ЦКИ редакции 1.3

DRIVER_CFG_*, DRIVER_*_CFG_SECTION

Маркеры названий разделов драйверов в конфигурационном файле ядра.

Формат:

```
#include <gis/gis_types.h>

#define DRIVER_S57_CFG_SECTION      "s57-local"
```

Библиотека:

gishelper

Описание:

Используются в качестве параметров функций данного раздела.

Наименования разделов

Идентификатор	Токен
DRIVER	"s57-local"
DRIVER	"sxf-local"
DRIVER	"shp-local"
DRIVER	"mtw-local"
DRIVER	"dted-local"
DRIVER	"geotiff-local"
DRIVER	"jpeg2000-local"
DRIVER	"png-local"
DRIVER	"rsw-local"

Наименования подразделов

Идентификатор	Токен
DRIVER	"acronym"
DRIVER	"fmt"
DRIVER	"secondary"
DRIVER	"alt"
DRIVER	"map"
DRIVER	"mapstyle"
DRIVER	"converter"

Примеры использования:

```

char opt_val[GIS_MAX_PATH_LENGTH];

if ( gis_core_config_get_driver_value( DRIVER_S57_CFG_SECTION,
                                      DRIVER_CFG_CONVERTER_PATH,
                                      &opt_val ) != EOK )
{
    printf( "gis_core_config_get_driver_value() failed!\n");
    return 1;
}

```

Тематические ссылки:

[gis_core_config_get_driver_value\(\)](#) [gis_core_config_get_value\(\)](#) [gis_core_config_set_driver_value\(\)](#)

Классификация:

ПК ЦКИ для ЗОСРВ «Нейтрино»

Поддерживается, начиная с ПК ЦКИ редакции 1.3

gis_helper_timer_get_result_sec()

Получение значения таймера.

Прототип:

```
#include <gis/gis_time.h>

double    gis_helper_timer_get_result_sec( gis_timer_t *timer );
```

Аргументы:

timer

Таймер.

Библиотека:

gishelper

Описание:

Функция возвращает разницу между отметками времени с точностью до микросекунд.

Примеры использования:

```
gis_timer_t      timer;

gis_helper_timer_start( &timer );

sleep( 1 );

gis_helper_timer_stop( &timer );

double time_passed = gis_helper_timer_get_result_sec( &timer );

if ( !time_passed )
{
    printf( "gis_helper_timer_get_result_sec() failed!\n");
    return 1;
}
```

Классификация:

ПК ЦКИ для ЗОСРВ «Нейтрино»

Безопасность использования	
<i>Обработчик прерываний</i>	Нет
<i>Обработчик сигналов</i>	Нет
<i>В потоке</i>	Да

Поддерживается, начиная с ПК ЦКИ редакции 1.3

Тематические ссылки:

`gis_timer_t`, `gis_helper_timer_start()`, `gis_helper_timer_stop()`

gis_helper_timer_start()

Запустить таймер.

Прототип:

```
#include <gis/gis_time.h>

void    gis_helper_timer_start( gis_timer_t *timer );
```

Аргументы:

timer

Таймер.

Библиотека:

gishelper

Описание:

Функция запускает таймер.

Примеры использования:

```
gis_timer_t      timer;

gis_helper_timer_start( &timer );
```

Классификация:

ПК ЦКИ для ЗОСРВ «Нейтрино»

Безопасность использования	
Обработчик прерываний	Нет
Обработчик сигналов	Нет
В потоке	Да

Поддерживается, начиная с ПК ЦКИ редакции 1.3

Тематические ссылки:

`gis_timer_t`, `gis_helper_timer_stop()`, `gis_helper_timer_get_result_sec()`

gis_helper_timer_stop()

Остановить таймер.

Прототип:

```
#include <gis/gis_time.h>

void    gis_helper_timer_stop( gis_timer_t *timer );
```

Аргументы:

timer

Таймер.

Библиотека:

gishelper

Описание:

Функция останавливает таймер.

Примеры использования:

```
gis_timer_t      timer;

gis_helper_timer_stop( &timer );
```

Классификация:

ПК ЦКИ для ЗОСРВ «Нейтрино»

Безопасность использования	
Обработчик прерываний	Нет
Обработчик сигналов	Нет
В потоке	Да

Поддерживается, начиная с ПК ЦКИ редакции 1.3

Тематические ссылки:

`gis_timer_t`, `gis_helper_timer_start()`, `gis_helper_timer_get_result_sec()`

gis_timer_t

Таймер.

Формат:

```
#include <gis/gis_time.h>

typedef struct
{
    struct timeval  t_start,
                  t_end;
} gis_timer_t;
```

Описание:

Структура определяет таймер.

Аргументы:

t_start

Начальная отметка времени.

t_end

Конечная отметка времени.

Классификация:

ПК ЦКИ для ЗОСРВ «Нейтрино»

Поддерживается, начиная с ПК ЦКИ редакции 1.3

Тематические ссылки:

[gis_helper_timer_start\(\)](#), [gis_helper_timer_stop\(\)](#), [gis_helper_timer_get_result_sec\(\)](#)

gis_kd_add()

Добавление нового узла.

Прототип:

```
#include <gis/gis_kdtree.h>

int gis_kd_add( gis_kd_tree_t *tree, double *coors, struct gis_kd_uniq_id *id);
```

Аргументы:

<i>tree</i>	Указатель на КД дерево
<i>coors</i>	Указатель на массив координат узла
<i>id</i>	Указатель на структуру идентификатора узла

Библиотека:

kd-tree

Описание:

Функция добавляет в дерево новый узел с координатами и идентификатором, переданными в качестве параметров.

Возвращаемое значение:

<i>0</i>	При добавлении узла произошла ошибка
<i>1</i>	Узел добавлен успешно

Примеры использования:

```
gis_kd_tree_t *kd_t;  
kd_t = gis_kd_init( 2 );  
  
struct gis_kd_uniq_id id;  
id.obj = NULL;  
id.attr = 1;  
  
double point[2] = { 10, 10 };  
if ( !gis_kd_add( kd_t, &point[0], &id ))  
{  
    printf( "KD-tree error: can't insert item" );  
    return 1;  
}
```

Классификация:

ПК ЦКИ для ЗОСРВ «Нейтрино»

Поддерживается, начиная с ПК ЦКИ редакции 1.0.0

gis_kd_destroy()

Удаление КД дерева.

Прототип:

```
#include <gis/gis_kdtree.h>

void gis_kd_destroy( gis_kd_tree_t *t );
```

Аргументы:

tree

Указатель на КД дерево

Описание:

Функция осуществляет корректное удаление дерева, освобождая выделенную под него память.

Примеры использования:

```
gis_kd_tree_t *kd_t;
kd_t = gis_kd_init( 2 );
gis_kd_destroy( kd_t );
```

Классификация:

ПК ЦКИ для ЗОСРВ «Нейтрино»

Поддерживается, начиная с ПК ЦКИ редакции 1.0.0

gis_kd_dnn()

Поиск узлов в заданном радиусе.

Прототип:

```
#include <gis/gis_kdtree.h>

int gis_kd_dnn( gis_kd_tree_t *tree, double *coors, double radius, struct gis_kd_uniq_id
↳ **ids, double **dists);
```

Аргументы:

tree

Указатель на КД дерево

coors

Указатель на массив координат узла

radius

Значение радиуса поиска

ids

Пустой массив идентификаторов для записи id найденных узлов

dists

Пустой массив расстояний для записи дистанции до найденных узлов

Библиотека:

kd-tree

Описание:

Функция осуществляет поиск узлов в заданном радиусе до точки, описанной массивом координат, переданных в качестве параметра coors. Идентификаторы найденных точек помещаются в массив ids, расстояния в массив dists.

Возвращаемое значение:

count

Количество найденных узлов

Примеры использования:

```

int count;
gis_kd_tree_t *kd_t;
kd_t = gis_kd_init( 2 );

struct gis_kd_uniq_id id;
double point[2];

id.obj = NULL;
id.attr = 1;
point[0] = 10;
point[1] = 10;
gis_kd_add( kd_t, &point[0], &id );

id.obj = NULL;
id.attr = 2;
point[0] = 20;
point[1] = 15;
gis_kd_add( kd_t, &point[0], &id );

point[0] = 19;
point[1] = 14;

struct gis_kd_uniq_id *found;
double *distances, radius = 3;

if ( ( count = gis_kd_dnn( kd_t, (double *)point, radius, &found, &distances ) ) == 0 )
{
    printf( "KD-tree: gis_kd_dnn found nothing" );
    return 1;
}

if ( found->attr != 2 )
{
    printf( "KD-tree found incorrect point\n" );
    return 1;
}

```

Классификация:

ПК ЦКИ для ЗОСРВ «Нейтрино»

Поддерживается, начиная с ПК ЦКИ редакции 1.0.0

gis_kd_init()

Создание КД дерева.

Прототип:

```
#include <gis/gis_kdtree.h>

gis_kd_tree_t* gis_kd_init(int dimensions);
```

Аргументы:

dimensions

Количество измерений (координат) у узлов дерева

Библиотека:

kd-tree

Описание:

Функция создает новое КД дерево для N измерений и возвращает указатель на него.

Возвращаемое значение:

!NULL

Указатель на созданное дерево

Примеры использования:

```
gis_kd_tree_t *kd_t;
kd_t = gis_kd_init( 2 );
gis_kd_destroy( kd_t );
```

Классификация:

ПК ЦКИ для ЗОСРВ «Нейтрино»

Поддерживается, начиная с ПК ЦКИ редакции 1.0.0

gis_kd_knn()

Поиск K ближайших узлов.

Прототип:

```
#include <gis/gis_kdtree.h>

int gis_kd_knn( gis_kd_tree_t *tree, int k, double *coors, struct gis_kd_uniq_id *ids,
↪ double *dists );
```

Аргументы:

tree

Указатель на КД дерево

k

Количество узлов, которое необходимо найти

coors

Указатель на массив координат узла

ids

Пустой массив идентификаторов для записи id найденных узлов

dists

Пустой массив расстояний для записи дистанции до найденных узлов

Библиотека:

kd-tree

Описание:

Функция осуществляет поиск ближайших узлов в дереве к точке, описанной массивом координат, переданных в качестве параметра coors. Идентификаторы найденных точек помещаются в массив ids, расстояния в массив dists.

Возвращаемое значение:

count

Количество найденных узлов

Примеры использования:

```

gis_kd_tree_t *kd_t;
kd_t = gis_kd_init( 2 );

struct gis_kd_uniq_id id;
double point[2];

id.obj = NULL;
id.attr = 1;
point[0] = 10;
point[1] = 10;
gis_kd_add( kd_t, &point[0], &id );

id.obj = NULL;
id.attr = 2;
point[0] = 20;
point[1] = 15;
gis_kd_add( kd_t, &point[0], &id );

point[0] = 19;
point[1] = 14;

struct gis_kd_uniq_id found[1];
double distances[1];

if ( gis_kd_knn( kd_t, 1, (double *)point, found, distances) != 1 )
{
    printf( "KD-tree error: can't find 2 nearest points" );
    return 1;
}

if ( found[0].attr != 2 )
{
    printf( "KD-tree found incorrect point\n" );
    return 1;
}

```

Классификация:

ПК ЦКИ для ЗОСРВ «Нейтрино»

Поддерживается, начиная с ПК ЦКИ редакции 1.0.0

gis_kd_rnn()

Поиск узлов в заданном диапазоне.

Прототип:

```
#include <gis/gis_kdtree.h>

int gis_kd_rnn( gis_kd_tree_t *tree, double *coors, struct gis_kd_uniq_id **ids);
```

Аргументы:

tree

Указатель на КД дерево

coors

Указатель на массив координат диапазона в N измерениях. Должны быть записаны в следующем формате:(мин1, мин2, ..., минN, макс1, макс2, ..., максN).

ids

Пустой массив идентификаторов для записи id найденных узлов

Библиотека:

kd-tree

Описание:

Функция осуществляет поиск узлов в заданном диапазоне, который задается минимальными и максимальными значениями через массив coors. Идентификаторы найденных точек помещаются в массив ids.

Возвращаемое значение:

count

Количество найденных узлов

Примеры использования:

```

int count;
gis_kd_tree_t *kd_t;
kd_t = gis_kd_init( 2 );

struct gis_kd_uniq_id id;
double point[2];

id.obj = NULL;
id.attr = 1;
point[0] = 10;
point[1] = 10;
gis_kd_add( kd_t, &point[0], &id );

id.obj = NULL;
id.attr = 2;
point[0] = 20;
point[1] = 15;
gis_kd_add( kd_t, &point[0], &id );

gis_kd_uniq_id *found;
double range[4];
range[0] = 14;
range[1] = 14;
range[2] = 21;
range[3] = 21;

if ( ( count = gis_kd_rnn( kd_t, (double *)range, &found )) == 0 )
{
    printf( "KD-tree error: gis_kd_dnn failed" );
    return 1;
}

if ( found->attr != 2 )
{
    printf( "KD-tree found incorrect point\n" );
    return 1;
}

```

Классификация:

ПК ЦКИ для ЗОСРВ «Нейтрино»

Поддерживается, начиная с ПК ЦКИ редакции 1.0.0

gis_kd_node_t

Структура узла дерева.

Формат:

```
#include <gis/gis_kdtree.h>

typedef struct Node gis_kd_node_t;
```

Описание:

Структура описывает узел КД дерева.

Классификация:

ПК ЦКИ для ЗОСРВ «Нейтрино»

Поддерживается, начиная с ПК ЦКИ редакции 1.0.0

gis_kd_tree_t

Структура КД дерева.

Формат:

```
#include <gis/gis_kdtree.h>

typedef struct Tree
{
    int          d_num;
    size_t       size;
    gis_kd_node_t *root;
} gis_kd_tree_t;
```

Описание:

Структура описывает КД дерево, поля интерпретируются следующим образом:

Аргументы:

<i>d_num</i>	Количество измерений (координат) у узлов дерева
<i>size</i>	Размер дерева (количество узлов)
<i>root</i>	Указатель на корень дерева

Примеры использования:

```
gis_kd_tree_t *kd_t;
kd_t = gis_kd_init( 2 );
```

Классификация:

ПК ЦКИ для ЗОСРВ «Нейтрино»

Поддерживается, начиная с ПК ЦКИ редакции 1.0.0

gis_kd_uniq_id

Структура идентификатора узла.

Формат:

```
#include <gis/gis_kdtree.h>

struct gis_kd_uniq_id
{
    void      *obj;
    uint64_t  attr;
};
```

Описание:

Структура описывает поля, которые используются для присвоения идентификаторов узлам КД дерева.

Аргументы:

obj

Указатель на объект

attr

Атрибут узла

Примеры использования:

```
struct gis_kd_uniq_id id;
id.obj = NULL;
id.attr = i;
if ( !gis_kd_add( kd_t, (double*)point, &id ))
{
    printf( "KD-tree error: can't insert item #%i\n", i );
    return 0;
}
```

Классификация:

ПК ЦКИ для ЗОСРВ «Нейтрино»

Поддерживается, начиная с ПК ЦКИ редакции 1.0.0

Работа с объектами

Поиск объектов, получение информации о них, а также их выделение

В разделе представлены примеры поиска, выделения объектов и отображения картографической информации. Ниже представлен код, реализующий поиск объекта. Сначала инициализируются параметры запроса (границы, пределы масштабов).

```
void MapWidget::showObjectInfo(QPointF pos)
{
    gis_core_request_parameters_t map;
    gis_core_request_parameters_init( &map );
    ...
}
```

Запрос списка объектов осуществляется с помощью класса `GisObjectList`. Класс содержит список объектов, отфильтрованных по заданным классам и границам карты, поэтому необходимо создать вектор класс-кодов.



Передача вектора с единственным значением класс-кода **GIS_CLASS_CODE_UNDEFINED** в конструктор `GisObjectList::GisObjectList()` осуществит выбор объектов любого класса.

```
void MapWidget::showObjectInfo(QPointF pos)
{
    gis_core_request_parameters_t map;
    gis_core_request_parameters_init( &map );

    std::vector<uint32_t> class_list;
    class_list.push_back(GIS_CLASS_CODE_UNDEFINED);
    GisObjectList list ( &map.borders, class_list );
    ...
}
```

Идентификация объекта осуществляется с помощью метода `GisObjectList::find_nearest_object()`, принимающий в качестве аргумента координаты точки. Метод находит ближайший объект к выделенной точке. Алгоритм поиска использует структуру k-мерных деревьев. Метод возвращает индекс объекта в списке.

```
void MapWidget::showObjectInfo(QPointF pos)
{
    ...
    object_point_t point_pos;
    /* Получение координат точки */
    ...
    int idx = list.find_nearest_object(point_pos);
    gis_object_t _obj;
    gis_object_init( &_obj );
    list.get_object(idx, _obj);
    ...
}
```

```
gis_object_free( &_obj );
}
```

На рисунке ниже представлен пример визуализации полученной информации о выделенном объекте с помощью представленного выше кода.

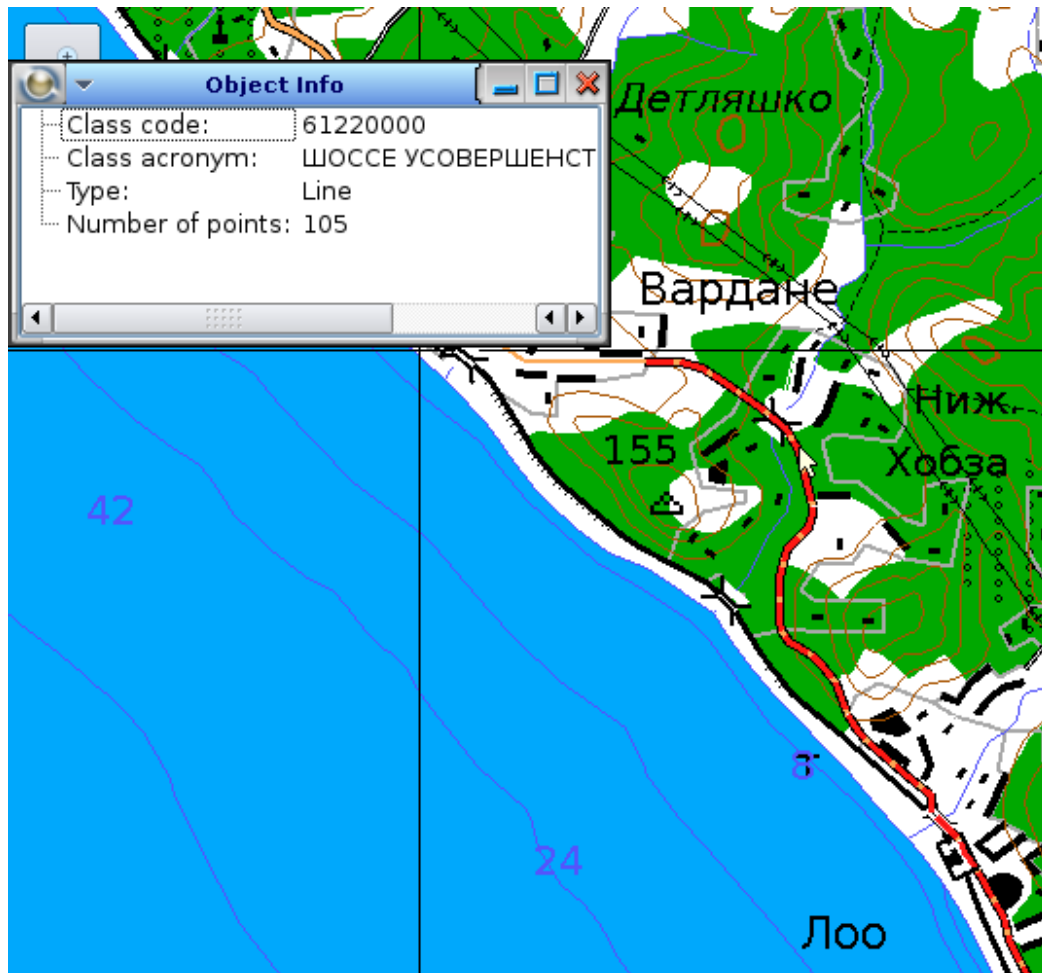


Рисунок 9. Просмотр информации об объекте.

Найденный объект можно выделить. Сделать это можно с помощью добавления пользовательского объекта. Пример создания пользовательского объекта:

```
void ObjectInfo::addSelectionLine( gis_object_t *obj )
{
    uint32_t color = 0xFF0000;
    uint32_t pen_width = 3;
    QVector <double_point_t> points;
    /*Заполнение вектора points точками*/
    ...
    selectionLine = gis_render_sm_userdata_add_polyline ( _sm_ctx, points.data(),
    ↪ obj->point_count, color, pen_width, 1 );
    gis_render_sm_redraw_userobject( _sm_ctx, true );
}
```

На рисунке ниже представлен пример выделения объекта.

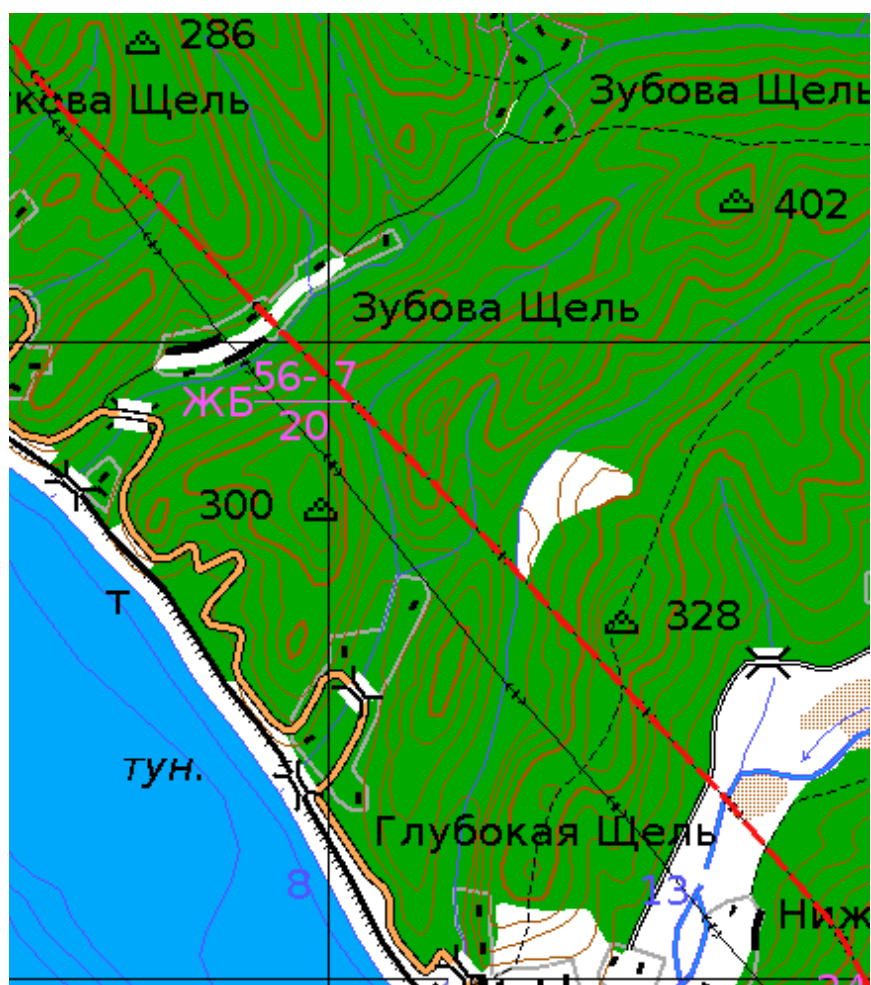


Рисунок 10. Выделение объекта.

Разработка драйверов источников данных

Разработка собственных драйверов источников данных на примере растрового формата BMP

Драйверы представляют собой часть менеджера ресурсов, отвечающую за работу с файлами определённых форматов.

Данная страница содержит:

- Описание доступных драйверов
- Особенности сборки драйверов
- Особенности проектирования драйверов
 - Структуры и макросы
 - Структура `gis_driver_ctx_t`
 - Структура `gis_driver_interface_t`
 - Функции
 - Прототипы требуемых функций
 - Примеры реализации требуемых функций
 - Функция `bmp_local_init`
 - Функция `bmp_local_convert_map`
 - Функция `bmp_local_connect`
 - Функция `bmp_local_get_connection_state`
 - Функция `bmp_local_get_info`
 - Функция `bmp_local_reset_file_cache`
 - Функция `bmp_local_synchronize_file_cache`
 - Функция `bmp_local_update_file_cache`

Описание доступных драйверов

s57-local

драйвер для работы с локальными электронными картами формата *S-57*;

sxf-local

драйвер для работы с локальными векторными картами формата *SXF*;

mtw-local

драйвер для работы с локальными растровыми картами формата *MTW*;

shp-local

драйвер для работы с локальными векторными картами формата *Shapefile*;

geotiff-local

драйвер для работы с локальными растровыми картами формата *GeoTIFF*;

jpeg2000-local

драйвер для работы с локальными растровыми картами формата *JPEG2000*;

dted-local

драйвер для работы с локальными растровыми картами формата *DTED*;

png-local

драйвер для работы с локальными растровыми картами формата *PNG*;

rsw-local

драйвер для работы с локальными растровыми картами формата *RSW*;

Особенности сборки драйверов

- Драйвер собирается по правилам сборки разделяемых библиотек (*dll*).
- Библиотека должна иметь имя **формат-local.so**. Например: **bmp-local.so**.
- Директория для инсталляции на целевой машине: **\$GIS_ROOT/sbin/drivers**.
- Необходимые пути для поиска библиотек и подключение общей библиотеки драйверов:

```
EXTRA_LIBVPATH += \$(INSTALL_ROOT_nto)/\$(CPUVARDIR)/opt/gis/lib
LIBS += drv_common
```

- Необходимые пути для поиска заголовочных файлов:

```
EXTRA_INCPATH += \$(INSTALL_ROOT_nto)/\$(CPUVARDIR)/opt/gis/include
```

Особенности проектирования драйверов



На данный момент протестировано добавление пользовательского драйвера только для растровых форматов, т.к. для них не требуется собственный конвертер (используется единый растровый).

Структуры и макросы

- Необходимые заголовочные файлы:

```
#include <gis/gis_time.h>
#include <gis/gishelper.h>
#include <gis/gis_driver.h>
#include <gis/gis_databuffer.h>
#include <gis/gis_core_cfg_parser.h>
```

- Необходимые макросы:

```
#define DRIVER_GCM_EXTENSION      ".local."GIS_GCM_FILE_EXTENSION
/*
 * Obligatory driver dependent defines
 */
#define DRIVER_ID                  GIS_CORE_DRIVERS_USER_R_LOCAL

#define DRIVER_CFG_SECTION         "bmp-local"

#define DRIVER_PREFIX              "[BMP_local_driver] "
```

- Идентификация драйвера приложениями [gis-monitor](#) и [gis-control](#) осуществляется при помощи [driver_id](#). Для каждого драйвера он задаётся макросом **DRIVER_ID**, возможные значения которого можно посмотреть на странице перечисления [gis_core_driver_id_t](#) в описании API. При необходимости создания пользовательского драйвера следует использовать `GIS_CORE_DRIVER_RS_USER_R_LOCAL`.
- Драйвер должен содержать структуры [gis_driver_interface_t](#) и [gis_driver_ctx_t](#).



См. [gis_driver_interface_t](#), [gis_driver_ctx_t](#)

Структура [gis_driver_ctx_t](#)

- Описание полей структуры:

Наименование сущности	Описание
<code>.connect</code>	Подключиться к серверу.
<code>.get_connection_state</code>	Получение статуса соединения с сервером.
<code>.get_info</code>	Получить путь до папки, содержащей кэш карт.
<code>.reset_cache</code>	Очистить источники и переэкшировать данные.
<code>.synchronize_cache</code>	Синхронизировать папки источника и внутреннего формата GCM. (Добавить недостающие карты и удалить те, источников которых не обнаружено.)
<code>.update_map_cache</code>	Перегенерировать карту во внутренний формат.

- Пример инициализации структуры:

```
gis_driver_interface_t driver_functions =
{
    .connect                = bmp_local_connect,
    .get_connection_state   = bmp_local_get_connection_state,
    .get_info               = bmp_local_get_info,
    .reset_cache            = bmp_local_reset_file_cache,
    .synchronize_cache      = bmp_local_synchronize_file_cache,
    .update_cache           = bmp_local_update_file_cache
};
```



См. ниже [требуемые функции](#).

Структура `gis_driver_interface_t`

- Описание полей структуры:

Наименование сущности	Описание
<code>.id</code>	Числовой идентификатор драйвера.
<code>.prefix</code>	Префикс драйвера для вывода отладочной информации.
<code>.initialized</code>	Флаг проверки инициализации драйвера.
<code>.state</code>	Флаг проверки установки соединения с ядром.
<code>.acronym</code>	Акроним драйвера.
<code>.mapstyle_dirname</code>	Имя директории хранения классификаторов/картоистилей.
<code>.origin_maps_dirname</code>	Имя директории в каталоге <code>\\$(GIS_CORE\↔_MAP_CACHE)</code> где хранятся исходные файлы карт.
<code>.origin_extension</code>	Расширение оригинального файла карты.
<code>.alt_origin_extensions</code>	Альтернативные расширения оригинального файла карты (если есть).
<code>.secondary_extensions</code>	Возможные расширения, если карта состоит из нескольких файлов.
<code>.driver_gcm_suffix</code>	Расширение файла после конвертации во внутренний формат.

- Пример инициализации структуры:

```
gis_driver_ctx_t driver_ctx =
{
    .id                = DRIVER_ID,
    .prefix             = DRIVER_PREFIX,

    .mapstyle_dirname   = NULL,
    .initialized         = false,
    .state              = NO_CONNECT,

    /* Conversion function is used as callback */
    .convert_map         = bmp_local_convert_map
};
```



Для заполнения других полей структуры `gis_driver_ctx_t` следует добавить требуемые записи в конфигурационном файле (`\gis-core.conf`)

```
<driver,name=bmp-local>
  <option,key=acronym>      BMP local</option>
  <option,key=fmt_extension> bmp</option>
  <option,key=map_folder>    BMP</option>
  <option,key=converter_path> sbin/gis-raster-processor</option>
</driver>
```

См. ниже [функция заполнения контекста из конфигурационного файла](#).

Функции

Прототипы требуемых функций

```
int bmp_local_convert_map( gis_core_map_information_t *map_info );
int bmp_local_connect( void );
int bmp_local_reset_file_cache( gis_core_user_data_t *udata );
int bmp_local_synchronize_file_cache( gis_core_user_data_t *udata );
int bmp_local_update_file_cache( int32_t map_id , gis_core_user_data_t *udata );
int bmp_local_get_info( gis_core_driver_info_t *info );
gis_core_connection_state_t bmp_local_get_connection_state( void );
```

Примеры реализации требуемых функций

Функция `bmp_local_init`

- При запуске ядра картографического сервиса с указанием списка драйверов при помощи опции **-d** считываемый аргумент опции используется для идентификации библиотеки и запуска инициализирующей функции. (Например: **-dbmp-local** сообщает ядру о том, что требуется загрузить библиотеку `bmp-local.so`).
- Функция инициализации драйвера должна иметь вид *имя_init*. Например: `bmp-local -> bmp_local_init`. Такая функция должна принимать в качестве аргумента строку опций вызова ядра и возвращать указатель на структуру `gis_driver_interface_t`.
- Внутри функции вызывается локальная функция заполнения контекста драйвера из конфигурационного файла:

```

static int get_driver_config_variables()
{
    /*Mandatory params*/
    int mandatory_error_flag = 0;

    if ( gis_common_driver_get_mandatory_driver_param_from_cfg(
↪ &driver_ctx.acronym,
                                                                    DRIVER_CFG_SECTION,
                                                                    DRIVER_CFG_ACRONYM,
                                                                    DRIVER_PREFIX ) != EOK
    ↪ )
    {
        mandatory_error_flag++;
    }
    if ( gis_common_driver_get_mandatory_driver_param_from_cfg(
↪ &driver_ctx.origin_extension,
                                                                    DRIVER_CFG_SECTION,
                                                                    DRIVER_CFG_FMT_EXTENSION,
                                                                    DRIVER_PREFIX ) != EOK
    ↪ )
    {
        mandatory_error_flag++;
    }

    /* extension + local.gcm */
    char *suffix = calloc( strlen( driver_ctx.origin_extension ) + strlen(
↪ DRIVER_GCM_EXTENSION ) + 1, sizeof( char ) );

    if ( !suffix )
    {
        gis_helper_debug_write( GIS_DEBUG_LEVEL_ERROR, DRIVER_PREFIX,
                                "Memory allocation error [%s()]", __FUNCTION__ );
        return ENOMEM;
    }

    strcpy( suffix, driver_ctx.origin_extension );

    driver_ctx.driver_gcm_suffix = strcat( suffix, DRIVER_GCM_EXTENSION );
    /* ----- */

    if ( gis_common_driver_get_mandatory_driver_param_from_cfg(
↪ &driver_ctx.origin_maps_dirname,
                                                                    DRIVER_CFG_SECTION,
                                                                    DRIVER_CFG_MAP_FOLDER,
                                                                    DRIVER_PREFIX ) != EOK
    ↪ )
    {
        mandatory_error_flag++;
    }
}

```

```

    if ( gis_common_driver_get_mandatory_driver_param_from_cfg(
↪ &driver_ctx.converter_path,
                                                    DRIVER_CFG_SECTION,
↪ DRIVER_CFG_CONVERTER_PATH,
                                                    DRIVER_PREFIX ) != EOK
↪ )
    {
        mandatory_error_flag++;
    }

    if ( mandatory_error_flag )
    {
        gis_helper_debug_write_lvl( GIS_DEBUG_LEVEL_ERROR, DRIVER_PREFIX,
↪ "Failed reading mandatory parameters from config [%s()]",
        __FUNCTION__ );
        return EINVAL;
    }

    return EOK;
}

```



См. `gis_common_driver_get_mandatory_driver_param_from_cfg()`

- Пример реализации основной функции:

```

enum driver_opts
{
    DRIVER_OPT_SYNC          = 0,
    DRIVER_OPT_END
};

static const char *driver_opts_keys[] =
{
    [DRIVER_OPT_SYNC]       = "sync",
    [DRIVER_OPT_END]        = NULL
};

static bool sync_initial_source      = false;
static bool sync_initial_source_is_hard = false;

gis_driver_interface_t *bmp_local_init( char *options )
{
    int config_return_res = get_driver_config_variables();
    if ( config_return_res != 0 )
    {

```

```

        gis_helper_debug_write_lvl( GIS_DEBUG_LEVEL_ERROR, DRIVER_PREFIX,
            "Failed to get config variables (RESULT=%d) [%s()]",
↪ config_return_res, __FUNCTION__ );
        return NULL;
    }

    char *value;

    while ( *options != '\0' )
        switch ( getsubopt( &options, ( char *const *__restrict )driver_opts_keys,
↪ &value ) )
        {
            case DRIVER_OPT_SYNC:
                sync_initial_source = true;
                if ( value == NULL )
                {
                    gis_helper_debug_write_lvl( GIS_DEBUG_LEVEL_INFO, DRIVER_PREFIX,
↪ "Source synchronization is enabled [%s()]", __FUNCTION__ );
                }
                else
                {
                    if ( !strcmp( value, "hard" ) )
                    {
                        sync_initial_source_is_hard = true;
                        gis_helper_debug_write_lvl( GIS_DEBUG_LEVEL_INFO,
↪ DRIVER_PREFIX, "Source hard synchronization is enabled [%s()]", __FUNCTION__ );
                    }
                    else
                    {
                        gis_helper_debug_write_lvl( GIS_DEBUG_LEVEL_INFO,
↪ DRIVER_PREFIX, "Source synchronization is enabled [%s()]", __FUNCTION__ );
                    }
                }
                break;

            default:
                gis_helper_debug_write_lvl( GIS_DEBUG_LEVEL_INFO, DRIVER_PREFIX,
↪ "Unknown option '%s' [%s()]", value, __FUNCTION__ );
                break;
        }

    if ( gis_common_driver_check_folders( &driver_ctx ) != EOK )
        return NULL;

    driver_ctx.initialized = true;

    if ( sync_initial_source )
    {
        if ( sync_initial_source_is_hard )
        {

```

```

        gis_common_driver_reset_gcm_database( driver_ctx.id );

        if ( bmp_local_reset_file_cache() != EOK )
            return NULL;
    }
    else
    {
        if ( bmp_local_synchronize_file_cache() != EOK )
            return NULL;
    }
}

return &driver_functions;
}

```



См. `gis_common_driver_check_folders`, `gis_common_driver_reset_gcm_database()`

Функция `bmp_local_convert_map`

- Пример реализации функции:

```

int bmp_local_convert_map( gis_core_map_information_t *map_info )
{
    gis_timer_t      timer;
    int              res = 0;
    const char       *directory      =
↳ gis_helper_env_get_maps_cache_directory();
    const char       *gis_root       = gis_helper_env_get_gis_root_directory();
    char              converter_path[GIS_MAX_PATH_LENGTH];
    char              map_filepath[GIS_MAX_PATH_LENGTH];
    char              gcm_filepath[GIS_MAX_PATH_LENGTH];
    char              orig_basename[GIS_MAX_PATH_LENGTH];

    gis_common_driver_get_basename_without_extension( map_info->src_fname,
↳ &orig_basename );

    gis_helper_timer_start( &timer );

    // Fill <gcm_fname> field as output
    sprintf( map_info->gcm_fname, "%s.%s", orig_basename,
↳ driver_ctx.driver_gcm_suffix );

    sprintf( map_filepath, "%s/%s/%s", directory, driver_ctx.origin_maps_dirname,
↳ map_info->src_fname );
    sprintf( gcm_filepath, "%s/%s/%s", directory, GIS_CACHE_SUBDIR_GCM,
↳ map_info->gcm_fname );

```

```

    sprintf( converter_path, "%s/%s", gis_root, driver_ctx.converter_path );

    gis_helper_debug_write_lvl( GIS_DEBUG_LEVEL_INFO, DRIVER_PREFIX, "Loading map
↪ '%s' to cache [%s()]", map_filepath, __FUNCTION__ );

    res = spawnl( P_WAIT, converter_path, converter_path,
                  "-m", map_filepath,
                  "-o", gcm_filepath,
                  NULL );

    if ( res != 0 ) {
        gis_helper_debug_write_lvl( GIS_DEBUG_LEVEL_ERROR, DRIVER_PREFIX,
↪ "Executing error: %d [%s()]", WEXITSTATUS( res ), __FUNCTION__ );
        return ( EFAULT );
    }

    gis_helper_timer_stop( &timer );
    gis_helper_debug_write_lvl( GIS_DEBUG_LEVEL_INFO, DRIVER_PREFIX, "bmp map
↪ conversion time: %f sec [%s()]", gis_helper_timer_get_result_sec( &timer ),
↪ __FUNCTION__ );

    return EOK;
}

```



См. `gis_common_driver_get_basename_without_extension`, `gis_helper_timer_start()`, `gis_helper_timer_stop()`

Функция `bmp_local_connect`

- Пример реализации функции:

```

int bmp_local_connect( void )
{
    if ( !driver_ctx.initialized )
        return ENXIO;                                // - Not initialized

    if ( driver_ctx.state == CONNECTED )
        return EALREADY;

    driver_ctx.state = CONNECTED;

    return EOK;
}

```

Функция `bmp_local_get_connection_state`

- Пример реализации функции:

```
gis_core_connection_state_t bmp_local_get_connection_state( void )
{
    return driver_ctx.state;
}
```

Функция `bmp_local_get_info`

- Пример реализации функции:

```
int bmp_local_get_info( gis_core_driver_info_t *info )
{
    const char *directory = gis_helper_env_get_maps_cache_directory();
    sprintf( info->orig_fpath, "%s/%s", directory, driver_ctx.origin_maps_dirname
↵ );
    return EOK;
}
```

Функция `bmp_local_reset_file_cache`

- Пример реализации функции:

```
int bmp_local_reset_file_cache( void )
{
    bmp_local_synchronize_file_cache();

    gis_helper_debug_write_lvl( GIS_DEBUG_LEVEL_INFO, DRIVER_PREFIX, "File cache
↵ database reset [%s()]", __FUNCTION__ );
    return EOK;
}
```

Функция `bmp_local_synchronize_file_cache`

- Пример реализации функции:

```
int bmp_local_synchronize_file_cache( void )
{
    return gis_common_driver_sync_cache( &driver_ctx, 0 );
}
```



См. `gis_common_driver_sync_cache()`

Функция `bmp_local_update_file_cache`

- Пример реализации функции:

```
int bmp_local_update_file_cache( int32_t map_id )
{
    if ( driver_ctx.state != CONNECTED )
        return ECONNREFUSED;

    return gis_common_driver_update_cache( &driver_ctx, map_id );
}
```



См. `gis_common_driver_update_cache()`

Безопасность использования

Дается представление об особенностях использования функций системных библиотек

Описание большей части функций системных библиотек снабжено специальной таблицей в параграфе "Классификация":

Безопасность использования	
Точка остановки потока	Нет
Обработчик прерываний	Да
Обработчик сигналов	Нет
В потоке	Нет

Таблица подобного вида раскрывает безопасность вызова в различных сценариях:

Точка остановки потока

Определяет возможность прерывания исполнения потока во время вызова функции (см. *pthread*_cancel()*).

Обработчик прерываний

Определяет возможность безопасного использования функции внутри обработчика прерывания (см. *InterruptAttach*).



Если функция небезопасна, то ее вызов из обработчика может привести к фатальным общесистемным последствиям. Последствия в общем случае непредсказуемы и могут включать аварийное завершение случайных процессов и даже процесса *procnto*-. В наилучшем сценарии подобного применения негаивный эффект проявится немедленно, в противном случае он будет проявляться в случайные моменты времени и связать его с конкретным обработчиком будет достаточно проблематично.

Обработчик сигналов

Безопасная функция обладает предсказуемым поведением как при вызове внутри обработчика сигналов, так и при прерывании сигналом другой небезопасной функции.

Некоторые из таких функций при неуспешном выполнении модифицируют *errno*, что необходимо учитывать при написании обработчика сигналов. Для этого следует отдельно сохранять (при входе в обработчик) и восстанавливать (при выходе) *errno*. В противном случае любой вызов функции, включая библиотечные, может получить асинхронно измененное значение *errno*, что в общем случае окажется непредсказуемым.

В потоке

Безопасность использования функции в многопоточных приложениях. Небезопасные функции могут иметь внутренние ресурсы, которые с некоторой вероятностью будут повреждены при ее одновременном вызове из нескольких потоков. Другие небезопасные вызовы могут иметь общий незащищенный ресурс в рамках своего класса функций. Такие функции в общем случае можно защитить отдельно, например, с использованием мьютекса.



Не является безопасным использование операций с плавающей точкой внутри обработчиков прерывания или сигналов.

Справочник по утилитам

Описываются информационные, диагностические и утилитарные компоненты комплекса и способы их использования

Справочник по утилитам содержит перечень приложений и утилит и рекомендации по их использованию.

Следующая таблица содержит информацию, которая поможет найти интересующую информацию:

Ссылка	Раздел документации
Приложения	Рекомендации по использованию приложений информационного сопровождения
Системные компоненты	Рекомендации по использованию основных системных компонентов комплекса
Утилиты	Рекомендации по использованию инструментальных компонентов

Типографические соглашения

Мы используем значки "заметки", "предостережения", и "предупреждения" для выделения важных сообщений:



"Заметки" указывают на нечто важное или полезное.



"Предостережения" сообщают о командах или процедурах, которые могут иметь нежелательные или побочные эффекты.



"Предупреждения" сообщают о командах или процедурах, которые могут быть опасны для Ваших файлов, данных, оборудования или даже Вас лично.

Вниманию пользователей Windows

В этой документации используется символ "прямой слэш" (/) в качестве разделителя во всех путях поиска, включая те, которые относятся к файлам в Windows.

Также в большинстве случаев мы следуем соглашениям о файловых системах в POSIX/UNIX.

Техническая поддержка

Для получения технической поддержки посетите раздел **Поддержка** сайта (www.kpda.ru). Вы обнаружите перечень предлагаемых способов оказания технической поддержки, включая публичный форум, форму обратной связи и контактные данные.

© 2017-2025, ООО «СВД ВС».

Приложения

Рекомендации по использованию приложений информационного сопровождения

Этот раздел предназначен для информирования о рекомендациях по использованию приложений информационного сопровождения.

Раздел включает:

- [Инструменты контроля и мониторинга](#)
- [Средства визуализации](#)

Инструменты контроля и мониторинга

Утилита	Описание
gis-monitor	Утилита для просмотра информации о доступных картах, а также управления конфигурацией ядра пакета.

Средства визуализации

Утилита	Описание
gis-filter-generator	Утилита позволяет создавать файл фильтрации классов, используемый в процессе конвертации карт.
gis-map-linker	Утилита для настройки ассоциативных связей векторных карт с конкретными стилями отображения.
gis-map-viewer	Утилита для просмотра визуальной картографической информации.
gis-raster-preview	Утилита просмотра растровой карты и основной информации о ней.
gis-rb-viewer	Минималистичное приложение для отображения картографической информации средствами библиотеки Render Buffer.

gis-monitor

Утилита для просмотра информации о доступных картах, а также управления конфигурацией ядра пакета.

Синтаксис:

```
gis-monitor [-xywhD]
```

Опции:

-x *position_x*

Опция задаёт позицию окна по горизонтали.

-y *position_y*

Опция задаёт позицию окна по вертикали.

-w *window_width*

Опция задаёт размер окна по горизонтали (ширину).

-h *window_height*

Опция задаёт размер окна по вертикали (высоту).

-D

Включить демонстрационный режим. При наличии этого флага, [gis-map-viewer](#), вызванный из `gis-monitor` будет работать в режиме выбранного драйвера. Например, только карта SXF без MTW.

Описание:

Приложение отображает общую информацию об имеющихся картах. А именно, для каждого драйвера выводится список карт, с дублированием в графическом виде. По каждой карте выводится справочная информация о границах, проекции.



Приложение взаимодействует с [gis-core](#).

Возвращаемое значение:

EXIT_SUCCESS

Успешное завершение.

EXIT_FAILURE

Утилита завершилась с ошибками.

Классификация:

ПК ЦКИ для ЗОСРВ «Нейтрино»

gis-filter-generator

Утилита позволяет создавать файл фильтрации классов, используемый в процессе конвертации карт.

Синтаксис:

```
gis-filter-generator -wh
```

Опции:

-w *window_width*

Опция задаёт размер окна по горизонтали (ширину).

-h *window_height*

Опция задаёт размер окна по вертикали (высоту).

Описание:

Утилита позволяет создавать файл фильтрации классов, используемый в процессе конвертации карт. При запуске прочитывает все доступные сконвертированные классификаторы из папки "gsm".

Возвращаемое значение:

0

Успешное завершение.

-1

Утилита завершилась с ошибками.

Классификация:

ПК ЦКИ для ЗОСРВ «Нейтрино»

gis-map-linker

Утилита для настройки ассоциативных связей векторных карт с конкретными стилями отображения.

Синтаксис:

```
gis-map-linker -wh
```

Опции:

-w *window_width*

Опция задаёт размер окна по горизонтали (ширину).

-h *window_height*

Опция задаёт размер окна по вертикали (высоту).

Описание:

Утилита позволяет настраивать ассоциативные связи векторных карт с конкретными стилями отображения.



Утилита взаимодействует с [gis-core](#).



Список доступных драйверов можно посмотреть на странице утилиты [gis-control](#).

Возвращаемое значение:

0

Успешное завершение.

-1

Утилита завершилась с ошибками.

Классификация:

ПК ЦКИ для ЗОСРВ «Нейтрино»

gis-map-viewer

Утилита для просмотра визуальной картографической информации.

Синтаксис:

```
gis-map-viewer [-xywhsD]
```

Опции:

- x lon_deg**
Опция задаёт градусную долготу отображения.
- y lat_deg**
Опция задаёт градусную широту отображения.
- w w_px**
Опция задаёт ширину виджета отображения карты.
- h h_px**
Опция задаёт высоту виджета отображения карты.
- s scale_idx**
Опция задаёт масштаб отображения карты.
- D driver_id**
Опция задаёт конкретный драйвер.

Описание:

Предназначена для отображения картографической информации и демонстрации возможностей API КПО ГИС . Программа позволяет управлять типами отображаемой информации, параметрами отображения и взаимодействовать с картографическим ядром.



Приложение взаимодействует с [gis-core](#).



Список доступных драйверов можно посмотреть на странице утилиты [gis-control](#).

Масштабы отображения карты:

Коэффициент масштаба	Физический масштаб
0	1:20 000 000
1	1:10 000 000
2	1:5 000 000
3	1:2 000 000
4	1:1 000 000
5	1:500 000
6	1:200 000
7	1:100 000
8	1:50 000
9	1:25 000
10	1:10 000
11	1:5 000
12	1:2 000

Возвращаемое значение:

EXIT_SUCCESS

Успешное завершение.

EXIT_FAILURE

Утилита завершилась с ошибками.

Классификация:

ПК ЦКИ для ЗОСРВ «Нейтрино»

gis-raster-preview

Утилита просмотра растровой карты и основной информации о ней.

Синтаксис:

```
gis-raster-preview -xyf
```

Опции:

-x *window_width*

Положение левого верхнего угла окна в пикселях по горизонтали.

-y *window_height*

Положение левого верхнего угла окна в пикселях по вертикали.

-f *filepath*

Путь до растровой карты.

Описание:

Приложение позволяет считывать и отображать растровые изображения из указанной растровой карты, включая все вложенные уменьшенные копии (если таковые имеются). Приложение также выводит информацию о географической точке привязки растра, размер изображения, географический масштаб растра.

Возвращаемое значение:

0

Успешное завершение.

1

Утилита завершилась с ошибками.

Классификация:

ПК ЦКИ для ЗОСРВ «Нейтрино»

gis-rb-viewer

Минималистичное приложение для отображения картографической информации средствами библиотеки *Render Buffer*.

Синтаксис:

```
gis-rb-viewer [-xywhs]
```

Опции:

-x *lon_deg*

Опция задаёт градусную долготу центра карты.

-y *lat_deg*

Опция задаёт градусную широту центра карты.

-w *w_px*

Опция задаёт ширину виджета.

-h *h_px*

Опция задаёт высоту виджета.

-s *scale_idx*

Опция задаёт индекс в таблице знаменателей масштабов отображения карты.

Описание:

Предназначена для отображения картографической информации и демонстрации возможностей API *Render Buffer*.

Масштабы отображения карты:

Коэффициент масштаба	Физический масштаб
0	1:20 000 000
1	1:10 000 000
2	1:5 000 000
3	1:2 000 000
4	1:1 000 000
5	1:500 000
6	1:200 000
7	1:100 000
8	1:50 000
9	1:25 000
10	1:10 000

11	1:5 000
12	1:2 000

Возвращаемое значение:

EXIT_SUCCESS

Успешное завершение.

EXIT_FAILURE

Утилита завершилась с ошибками.

Классификация:

ПК ЦКИ для ЗОСРВ «Нейтрино»

Поддерживается, начиная с ПК ЦКИ редакции 1.2

Системные компоненты

Рекомендации по использованию основных системных компонентов комплекса

Этот раздел предназначен для информирования о рекомендациях по использованию основных системных компонентов комплекса.

Раздел включает:

- [Перечень системных компонентов](#)

Перечень системных компонентов

Утилита	Описание
gis-core	Исполняемый файл картографического сервиса.
gis-raster-processor	Конвертер растровых карт во внутренний формат.
gis-s57-processor	Конвертер карт s57 во внутренний формат.
gis-shape-processor	Конвертер карт Shapefile во внутренний формат.
gis-sxf-processor	Конвертер карт sxf во внутренний формат.

Исполняемый файл картографического сервиса.

Синтаксис:

```
gis-core [-fd]
```

Опции:

-f

Опция включает режим экономии разделяемой памяти (приоритет отдается файловым операциям).

-d driver,opt1=val1,opt2=val2

Опция отвечает за подгрузку драйверов источников данных (аргумент опции *driver* задает имя подгружаемого драйвера, полный их перечень представлен ниже). Опция может быть задана повторно. Параметры запуска конкретного драйвера задаются через запятую после указания имени драйвера.

-v

Вывод отладочной информации (сведения о версионировании и размеры служебных полей).

Описание:

Данное приложение является основным компонентом картографического фреймворка и составляет картографический сервис. Подавляющее большинство библиотек, утилит и приложений тем или иным образом взаимодействуют с данным сервисом и требуют его предварительного запуска.

При передаче приложению опции **-f** сервис стремится минимизировать потребление оперативной памяти для хранения картографической информации, отдавая предпочтение файловым операциям с картографическим кэшем. В противном случае ядро картографического сервиса использует объекты разделяемой памяти для накопления картографической информации и снижения накладных расходов по доступу к ней.

Для успешного исполнения сервиса требуется активация хотя бы одного драйвера источника данных. Это осуществимо путем передачи одной или нескольких опций **-d driver**, где *driver* определяет имя загружаемого модуля драйвера. Перечень поддерживаемых драйверов, а также их опции представлены ниже в соответствующем разделе. Параметры запуска конкретного драйвера могут быть переданы как через опцию **-d**, так и в конфигурационном файле `$GIS_CONFIG_DIRECTORY/gis-core.conf`. Через опцию **-d** параметры драйверов определяются следующим образом: **-d driver,opt1=val1,opt2=val2**.

Поддерживаемые драйвера источников данных:

Драйвера локальных источников данных:

Драйвер	Директория кэша карт
s57-local	<code>\$GIS_CORE_MAP_CACHE/S-57</code>
sxf-local	<code>\$GIS_CORE_MAP_CACHE/sxf</code>

mtw-local	\$GIS_CORE_MAP_CACHE/mtw
shp-local	\$GIS_CORE_MAP_CACHE/Shapefile
geotiff-local	\$GIS_CORE_MAP_CACHE/GeoTIFF
jpeg2000-local	\$GIS_CORE_MAP_CACHE/JPEG2000
dted-local	\$GIS_CORE_MAP_CACHE/DTED
png-local	\$GIS_CORE_MAP_CACHE/PNG
rsw-local	\$GIS_CORE_MAP_CACHE/RSW

Для всех вышеуказанных драйверов локальных источников данных поддерживаются следующие опции синхронизации кэша карт в момент запуска сервиса:

Опция драйвера	Комментарий
sync=soft	В директории кэша карт осуществляется лишь отслеживание изменений.
sync=hard	Производится полная перезапись кэша карт.

Драйвера удалённых источников данных:

- **kks-server**

Директория кэша карт: \$GIS_CORE_MAP_CACHE/kks

Для данного драйвера поддерживаются следующие опции:

Опция драйвера	Комментарий
ip=<server_IP>	IP адрес сервера ККС.
port=<server_port>	Удаленный порт сервера ККС.
sync=<soft/hard>	Принудительная синхронизация кэша карт в момент запуска сервиса. В режиме soft осуществляется лишь отслеживание изменений в директории драйверного кэша карт. В режиме hard производится полная перезапись кэша карт.

- **rer-server**

Директория кэша карт: \$GIS_CORE_MAP_CACHE/rer

Для данного драйвера поддерживаются следующие опции синхронизации кэша карт в момент запуска сервиса:

Опция драйвера	Комментарий
sync=soft	В директории кэша карт осуществляется лишь отслеживание изменений.
sync=hard	Производится полная перезапись кэша карт.
sync=imit	Происходит включение имитатора, драйвер работает только с локальными данными (без сетевого взаимодействия).

- **uks-server**

Директория кэша карт: \$GIS_CORE_MAP_CACHE/uks

Для данного драйвера поддерживаются следующие опции синхронизации кэша карт в момент запуска сервиса:

Опция драйвера	Комментарий
sync=soft	В директории кэша карт осуществляется лишь отслеживание изменений.

sync=hard	Производится полная перезапись кэша карт.
sync=imit	Происходит включение имитатора, драйвер работает только с локальными данными (без сетевого взаимодействия).

Возвращаемое значение:

0

Успешное завершение.

1

Утилита завершилась с ошибками.

Классификация:

ПК ЦКИ для ЗОСРВ «Нейтрино»

gis-raster-processor

Конвертер растровых карт во внутренний формат.

Синтаксис:

```
gis-raster-processor -m oc
```

Опции:

-m

Путь до входного файла растровой карты.

-o

Путь до выходного файла карты в формате GCM.

-c

Требуемые параметры проекции растра, используемые при перепроецировании. Формат аргумента - EPSG-код, PROJ-строка (поддерживаемые gdalwarp типы. Например "ESRI:54003", "+proj=utm +ellps=WGS84 +zone=35"). По умолчанию, перепроецированный растр временно помещается в директорию исходного растра и имеет в имени файла префикс *reproj_*. Для задания директории временного хранения перепроецированных растров используется переменная окружения **GIS_RASTER_PROCESSOR_TMP_DIR**. Задание этой переменной окружения удобно в случае, если исходный растр находится на накопителе с малой скоростью записи и/или имеет ограниченный объём.

Описание:

Данное приложение является одним из компонентов картографического фреймворка. Приложение конвертирует растровые карты во внутренний gcm формат, поддерживает блочную структуру растровых данных, работает с матрицами высот и трёхканальными RGB-растрами, а также с их уменьшенными копиями.

Возвращаемое значение:

EOK

Успешное завершение.

EINVAL

Некорректные входные параметры.

ENOMEM

Не удалось выделить память.

EBADF

Не удалось открыть/прочитать входной файл.

EOverflow

Координаты в градусах находятся вне диапазона.

ENOTSUP

Неизвестная ошибка.

ENODATA

Нет объектов на карте.

EBADFD

Не удалось открыть/прочитать GCM файл.

EFAULT

Не удалось сконвертировать.

Классификация:

ПК ЦКИ для ЗОСРВ «Нейтрино»

Поддерживается, начиная с ПК ЦКИ редакции 1.1.0

gis-s57-processor

Конвертер карт s57 во внутренний формат.

Синтаксис:

```
gis-s57-processor -mofg
```

Опции:

-m map

Опция определяет имя файла. Файл должен являться картой формата S57.

-o gsm

Опция задает имя выходного файла в формате gsm.

-c projection

Опция определяет проекцию. Проекция передаётся в виде строки.

-f filter_file

Опция определяет имя файла фильтрации классов.

-g clipping_scale

Опция задействует нарезку полученной карты на листы по номенклатурной сетке с переданным масштабом.

Описание:

Приложение конвертирует карты формата s57 во внутренний gsm формат. Использование формата необходимо для других картографических утилит, например, для отрисовки карты.

Исходный файл формата S57 должен иметь расширение .000 Выходному файлу внутреннего формата gsm рекомендуется указывать расширение .gsm

Возвращаемое значение:

0

Успешное завершение.

1

Утилита завершилась с ошибками.

Классификация:

ПК ЦКИ для ЗОСРВ «Нейтрино»

gis-shape-processor

Конвертер карт Shapefile во внутренний формат.

Синтаксис:

```
gis-shape-processor -mofg
```

Опции:

-m

Путь к папке, содержащей конвертируемую карту формата Shapefile. В папке должен быть файл стиля отображения style.sld и параметры карты meta.info

-o

Путь к выходному GCM файлу.

-c

Опция определяет проекцию. Проекция передается в виде строки.

-f filter_file

Опция определяет имя файла фильтрации классов.

-g clipping_scale

Опция задействует нарезку полученной карты на листы по номенклатурной сетке с переданным масштабом.

Описание:

Приложение конвертирует карты формата Shapefile во внутренний gcm формат.

Возвращаемое значение:

EOK

Успешное завершение.

EINVAL

Некорректные входные параметры.

ENOMEM

Не удалось выделить память.

EBADF

Не удалось открыть/прочитать shape файл.

EBADFD

Не удалось открыть/прочитать GCM файл.

EFAULT

Не удалось сконвертировать.

Классификация:

ПК ЦКИ для ЗОСРВ «Нейтрино»

gis-sxf-processor

Конвертер карт *sxf* во внутренний формат.

Синтаксис:

```
gis-sxf-processor -dmRrocfg
```

Опции:

- d**
Переконвертировать все классификаторы, которые находятся в стандартной директории.
- m file**
Путь к файлу карты SXF.
- R dir**
Путь к пользовательской директории RSC.
- r file**
Путь к классификатору, который необходимо использовать.
- o file**
Путь, по которому должен лежать выходной GCM файл.
- c**
ID оффлайн конвертации проекции.
- f filter_file**
Опция определяет файл фильтрации классов.
- g clipping_scale**
Опция знаменатель масштаба для нарезки карты на номенклатурные листы.

Описание:

Данное приложение является одним из компонентов картографического фреймворка. Приложение конвертирует карты формата *sxf* во внутренний *gsm* формат. Использование формата необходимо для других картографических утилит, например, для отрисовки карты.

Возвращаемое значение:

- EOK**
Успешное завершение.
- EINVAL**
Некорректные входные параметры.
- ENOMEM**
Не удалось выделить память.

EBADF

Не удалось открыть/прочитать SXF файл.

EBADR

Не удалось найти/открыть/прочитать RSC файл.

EOVERFLOW

Координаты в градусах находятся вне диапазона.

ENOTSUP

Неизвестная ошибка.

ENODATA

Нет объектов на карте.

EBADFD

Не удалось открыть/прочитать GCM файл.

EFAULT

Не удалось сконвертировать.

Классификация:

ПК ЦКИ для ЗОСРВ «Нейтрино»

Утилиты

Рекомендации по использованию инструментальных компонентов

Этот раздел предназначен для информирования о рекомендациях по использованию инструментальных компонентов.

Раздел включает:

- Утилиты общего назначения
- Утилиты для работы с картами S-63, S-57
- Утилиты для работы с растровыми данными

Утилиты общего назначения

Утилита	Описание
gis-buffer-renderer	Утилита для рендеринга картографической информации в память.
gis-control	Утилита контроля картографического сервиса, кэша карт и взаимодействия с источниками информации.
gis-gcm-clip	Утилита для нарезки карт формата gcm на номенклатурные листы.
gis-gcm-info	Утилита отображения заголовка GCM файла.
gis-mapstyle-info	Утилита просмотра информации о картостиле.

Утилиты для работы с картами S-63, S-57

Утилита	Описание
gis-s63-decrypt	Декодер карт формата S-63/S-57 в ZIP архив.
gis-s63-gen-userpermit	Генератор UserPermit по заданным M_KEY, M_ID, HW_ID.

gis-s63-unpack	Утилита для распаковки карт, созданных по стандарту S-63/S-57.
----------------	--

Утилиты для работы с растровыми данными

Утилита	Описание
gis-rasterizer	Растеризатор векторных карт.

gis-buffer-renderer

Утилита для рендеринга картографической информации в память.

Синтаксис:

```
gis-buffer-renderer [-whxysdFufoe]
```



Обязательные опции.

Опции:

-w *w_px*

Опция задаёт ширину изображения карты.

-h *h_px*

Опция задаёт высоту изображения карты.

-x *lon_deg*

Опция задаёт градусную долготу центра изображения.

-y *lat_deg*

Опция задаёт градусную широту центра изображения.

-s *scale_denominator*

Опция задаёт знаменатель масштаба изображения карты.

-d *sharedMemoryId*

Идентификатор разделяемой области памяти.



Следующие опции необходимы при рисовании в файл.

-F *resultFilePath*

Путь к файлу, в котором будет сохранено результирующее изображение.



Следующие опции необходимы при работе с сервером.

-u server_url

Адрес сервера, на который отправляется изображение (при условии работы в корректном режиме).

-f image_format

Формат изображения: bmp, jpg, png.

-o order_id

Идентификатор заказа.

-e errorCode

Устанавливаемый сервером код для оповещения об ошибке.

Описание:

Утилита предназначена для отрисовки картографической информации.

Возвращаемое значение:

EXIT_SUCCESS

Успешное завершение.

EXIT_FAILURE

Утилита завершилась с ошибками.

Классификация:

ПК ЦКИ для ЗОСРВ «Нейтрино»

Поддерживается, начиная с ПК ЦКИ редакции 1.2

gis-control

Утилита контроля картографического сервиса, кэша карт и взаимодействия с источниками информации.

Синтаксис:

```
gis-control [-ludrvigs]
```

Опции:

- l**
Получение списка драйверов, с которыми запущен [gis-core](#).
- u**
Обновление метаинформации каталога карт в сервисе без изменения файлов карт (см. описание [GIS_CORE_UPDATE_CACHE_MODE_SOFT_UPDATE](#)).
- d DriverIndex**
Опция задает индекс драйвера для вывода подробной информации о доступных картах. Для вывода информации по всем доступным драйверам необходимо передать параметр "ALL".
- r**
Полное удаление каталога карт GCM и создание новых из оригинального источника (см. описание [GIS_CORE_UPDATE_CACHE_MODE_HARD_RESET](#)).
- v**
Уровень подробности вывода.
- i**
Вывести информацию о версии и ревизии программного продукта.
- g**
Запрос на загрузку карт.
- s sharedMemoryId**
Запрос на заполнение разделяемой области памяти всеми доступными картами по переданному идентификатору.

Описание:

Данная утилита может быть использована для контроля картографического сервиса, кэша карт и взаимодействия с источниками информации.

Утилита позволяет получить список доступных драйверов, обновить и переписать кэш карт, а также получить информацию о доступных картах в табличном виде.

Существует возможность выбрать уровень подробности вывода табличной информации с помощью опции -v. Без использования опции выводится информация о номере, имени, масштабе, проекции и эллипсоиде карт. При однократном использовании опции добавляются колонки с координатами, количеством классов и объектов. При двукратном использовании опции выводится детализация по классам объектов в картах (код и акроним класса, количество объектов).



Утилита взаимодействует с [gis-core](#).

Доступные драйвера:

```
gis-control -l

0   Local S57
1   Local SXF
2   Local MTW
3   Local SHP
4   Local GeoTIFF
5   Local JPEG2000
6   Local DTED
7   Local PNG
8   Local RSW
9   RER Server
10  UKS Server
11  KKS Server
```

Возвращаемое значение:

- 0** Успешное завершение.
- 1** Утилита завершилась с ошибками.
- 2** Подключение к [gis-core](#) неудачно.

Классификация:

ПК ЦКИ для ЗОСРВ «Нейтрино»

gis-gcm-clip

Утилита для нарезки карт формата gcm на номенклатурные листы.

Синтаксис:

```
gis-gcm-clip [-ios]
```

Опции:

-i input

Путь к исходному файлу для нарезки.

-o output

Путь до выходной директории, в которой будут сформированы нарезанные файлы (раскладываются по папкам с названием номенклатурного листа).

-s scale

Масштаб, согласно которому должна осуществляться нарезка. Поддерживаемые значения: 1 000 000, 100 000, 50 000, 25 000

Описание:

Утилита позволяет нарезать исходную карту формата gcm произвольного размера на номенклатурные листы заданного масштаба.

Возвращаемое значение:

0

Успешное завершение.

1

Утилита завершилась с ошибками.

Классификация:

ПК ЦКИ для ЗОСРВ «Нейтрино»

gis-gcm-info

Утилита отображения заголовка GCM файла.

Синтаксис:

```
gis-gcm-info [-f] /path/to/gcm
```

Опции:

-f Путь к файлу GCM.

Описание:

Данная утилита может быть использована для отображения основной информации о GCM файле карты.

Возвращаемое значение:

0 Успешное завершение.

1 Утилита завершилась с ошибками.

Классификация:

ПК ЦКИ для ЗОСРВ «Нейтрино»

gis-mapstyle-info

Утилита просмотра информации о картостиле.

Синтаксис:

```
gis-mapstyle-info [-ste]
```

Опции:

-s filename

Опция определяет файл картостиля. Аргумент filename задает имя (путь) файла.

-t type

Опция задает тип картостиля. Допустимы следующие значения:

- RSC — картостиль в формате RSC (в исходной терминологии - классификатор), обычно применяемый с картами в формате SXF.
- SLD — картостиль в формате SLD, обычно применяемый с картами в формате SHP.
- S52 — картостиль в формате S-52, обычно применяемый с картами в формате S-57.

-e command

Опция позволяет определить выводимую информацию о картостиле. В зависимости от выбранного type перечень доступных для исполнения команд различается.

Описание:

Приложение позволяет получать различную информацию о картостиле. В зависимости от значения аргумента type утилита будет интерпретировать файл картостиля filename в качестве одного из поддерживаемых форматов. При этом для каждого формата предусмотрен собственный перечень поддерживаемых команд (опция -e command), позволяющих отображать различные сведения о картостиле.

Поддерживаемые команды в зависимости от типа картостиля:

Команда	Тип классификатора	Описание
---------	--------------------	----------

images	RSC	<p>Утилита создает в текущей директории каталоги ./data/image и сохраняет извлеченные из классификатора векторные и растровые изображения символов в форматах SVG и PNG соответственно. Названия результирующих файлов имеют вид classcode.imagetype, где:</p> <ul style="list-style-type: none"> • classcode — номер класса объекта, содержащего изображение • imagetype — расширение svg или png, выбранное в зависимости от типа изображения <p>Если файл с именем classcode.←imagetype уже существует, добавляется суффикс (n), обозначающий номер повтора.</p>
layer_count	RSC, SLD	Утилита выводит информацию о количестве слоев в картостиле.
layer_list	RSC, SLD	Утилита выводит список наименований для слоев картостиля.
class_count	ALL	Утилита выводит информацию о количестве классов в картостиле.
class_list	SLD, S52	Утилита выводит список классов картостиля и их наименований.
info	ALL	Утилита выводит основную информацию о картостиле.

Возвращаемое значение:

0

Успешное завершение.

1

Утилита завершилась с ошибками.

Классификация:

ПК ЦКИ для ЗОСРВ «Нейтрино»

gis-s63-decrypt

Декодер карт формата S-63/S-57 в ZIP архив.

Синтаксис:

```
gis-s63-decrypt -NC [-HUKPM]
```

Опции:

-N *FileName*

Опция определяет исходный файл формата S-57, защищенный по стандарту S-63. Файл должен иметь расширение ".000"]

-C *CellPermit*

Опция задает Cell Permit для исходного файла.

-H *HW_ID*

Опция задает HW_ID в формате ASCII.

-U *UserPermit*

Опция задает UserPermit. Используется в паре с опцией -K.

-K *M_KEY*

Опция задает M_KEY. Используется в паре с опцией -U.

-P *ProcNum*

Опция активирует режим поиска HW_ID и задает количество процессоров.

-M *Mode*

Опция определяет режим поиска, "D" - перебор цифровых HW_ID, "L" - перебор HW_ID с наличием букв.Используется в паре с опцией -P.

Описание:

Данная утилита конвертирует закодированные файлы карт формата S-63/S-57 с расширением ".000". Результирующий файл в соответствии со стандартом S-63 должен являться ZIP архивом.

Утилита работает как с помощью напрямую заданного HW_ID, так и на основе пары значений M_KEY и UserPermit. При отсутствии этих данных утилита может осуществить поиск подходящего HW_ID. Поиск активируется опцией -P, значение опции определяет количество процессов, которые будут одновременно искать подходящий HW_ID. Рекомендуется в качестве данного значения задавать количество ядер процессора на используемой технике.

Существует возможность выбрать режим поиска с помощью опции **-M**. Режим "D" определяет перебор HW_ID, которые содержат в себе исключительно цифры в диапазоне 0-9, режим "L" задает перебор остальных вариантов из словаря 0-F, за исключением диапазона значений, определенного опцией **-M D**. Если режим перебора не задан, утилита осуществляет полный поиск по всему словарю.

Возвращаемое значение:

0

Успешное завершение.

1

Утилита завершилась с ошибками.

2

Поиск HW_ID завершился неудачно.

Классификация:

ПК ЦКИ для ЗОСРВ «Нейтрино»

gis-s63-gen-userpermit

Генератор UserPermit по заданным *M_KEY*, *M_ID*, *HW_ID*.

Синтаксис:

```
gis-s63-gen-userpermit -KMH
```

Опции:

-K M_KEY

Опция задает значение *M_KEY* в форме строки, состоящей из 5 hex-символов.

-M M_ID

Опция задает значение *M_ID* в формате ASCII (4 цифры).

-H HW_ID

Опция задает значение *HW_ID* в форме строки, состоящей из 5 hex-символов.

Описание:

Данная утилита генерирует UserPermit по заданным *M_KEY*, *M_ID*, *HW_ID*.

Используя *M_KEY* утилита на основе алгоритма Blowfish кодирует *HW_ID*, добавляет к строке с результатом полученную с помощью алгоритма CRC32 контрольную сумму и *M_ID*. Итоговая строка представляет собой сгенерированный UserPermit и выводится на экран.

Возвращаемое значение:

0

Успешное завершение.

1

Утилита завершилась с ошибками.

Классификация:

ПК ЦКИ для ЗОСРВ «Нейтрино»

gis-s63-unpack

Утилита для распаковки карт, созданных по стандарту S-63/S-57.

Синтаксис:

```
gis-s63-unpack <cache_dir> <S57_dir> <logfile> <hw_id> <u_opt> <m_key>
```

Опции:

cache_dir

Директория, содержащая кэш карт.

S57_dir

Директория, содержащая карты формата S-57.

logfile

Задаёт путь к файлу, куда будут сохранены логи работы утилиты.

hw_id

Задаёт уникальный идентификатор аппаратного устройства HW_ID.

u_opt

Задаёт userpermit, уникальные для пользователя права доступа.

m_key

Задаёт уникальный идентификационный ключ M_KEY.

Описание:

Утилита позволяет распаковать карты из *cache_dir* в директорию *S57_dir*. Для распаковки необходимо указать соответствующие HW_ID, M_KEY и User Permit. Процесс раскодирования выполняется с помощью утилиты [gis-s63-decrypt](#). Скрипт проходится по директории и вызывает для всех карт формата S-57 [gis-s63-decrypt](#), который работает для одного файла карты и раскодирует ее.

Возвращаемое значение:

0

Успешное завершение.

1

Ошибка распаковки.

Классификация:

ПК ЦКИ для ЗОСРВ «Нейтрино»

gis-rasterizer

Растеризатор векторных карт.

Синтаксис:

```
gis-rasterizer [-asco] source_file
```

Опции:

- a** Атрибут(-ы), по которому(-ым) требуется отбирать объекты для растеризации. Можно указать произвольное количество атрибутов, при помощи нескольких параметров -a.
- s** Размер пикселя в метрах для выходного растра.
- c** Путь к файлу классификатора исходной векторной карты.
- o** Путь к выходному файлу растра. Расширение файла указывать не обязательно.
- file** Путь к файлу входной векторной карты.



Для форматов SXF и S-57 в параметре **file** следует указывать непосредственно файл карты, в то время как для формата Shapefile требуется указывать путь к директории, содержащую файлы карты (с расширением .shp).

Описание:

Данная утилита предназначена для генерации растров (матриц высот, DEM) формата GeoTIFF из векторных форматов карт. Утилита позволяет пользователю задать атрибут, по которому будет производиться отбор объектов для помещения в растр (например, абсолютная высота), установить размер выходного изображения при помощи параметра **-s**. Результирующий растр размещается по пути, указанному в параметре **-o**.

Поддерживаемые форматы векторных карт:

Формат	Описание
SXF	Векторная карта в формате хранения и обмена SXF (Storage and eXchange Format).
S-57	Векторная карта в формате S-57 (морская картографическая информация).
Shapefile	Векторная карта в формате Shapefile (директория, содержащая файлы .shp).

Доступные выходные форматы:

Формат	Описание
GeoTIFF	Одноканальный растр (матрица высот).

Возвращаемое значение:

EOK

Успешное завершение.

EINVAL

Некорректные входные параметры.

ENOMEM

Не удалось инициализировать внутренние структуры для работы с файлами.

EBADF

Не удалось открыть исходный файл или создать выходной файл.

ENOTSUP

Не удалось найти атрибуты высоты (рельефа) в исходной карте.

ENODATA

Не удалось установить проекцию для выходного растра.

EFAULT

Не удалось растеризовать объекты векторной карты.

EIO

Не удалось сохранить выходной растр.

Классификация:

ПК ЦКИ для ЗОСРВ «Нейтрино»