

Пакет СТХ\_АРІ. Описание, процедуры и функции  
Версия 1.0.0.5

Введение
----------

Программный интерфейс CTX\_API предназначен для обработки поисковых запросов к системе поиска на основе Oracle Text. Данный интерфейс использует **иерархический тезаурус русского языка**<sup>1</sup>. Он позволяет как полностью обрабатывать поисковый запрос, так и выполнять различные функции более низкого уровня, вызываемые, например, из интерфейса приложения (реализующего функции расширенного поиска итп).

Интерфейс написан на языке PL/SQL и реализован также в виде WSDL-сервиса, использующего стандартный протокол SOAP для организации взаимодействия B2B<sup>2</sup>.

CTX\_API реализует функцию высокоскоростного универсального парсера и вспомогательных вызовов пакета CTXSYS.CTX\_THES. Основной функцией интерфейса является выполнение интеллектуального парсинга поисковых строк (запросов) на основе метаданных загруженных тезаурусов для последующего выполнения запроса поисковым движком Oracle Text<sup>3</sup>.

Специфика интерфейса CTX\_API такова, что он не привязан к использованию одного конкретного естественного языка. Пакет способен работать с любым языком, использующим слова, разделяемые пробелами. Это делает возможным одновременное использование любого количества тезаурусов<sup>4</sup>.

Пакет не привязан к той или иной целевой аппаратно-программной платформе и может быть использован на всех платформах, поддерживаемых Oracle.

Поддерживаемые версии СУБД Oracle: > 9.2.0.3<sup>5</sup>

---

<sup>1</sup> Может использовать любое количество иерархических тезаурусов на любом языке. Парсер поисковых запросов способен функционировать и при отсутствии загруженных тезаурусов (в режиме KEYWORD).

<sup>2</sup> B2B – Business-To-Business. Общепринятая аббревиатура, обозначающая взаимодействие типа «Сервис-сервис». Подразумевается предоставление сервисных услуг поверх глобальной сети одного бизнеса другому.

<sup>3</sup> И других, выполненных в соответствие со стандартами ISO-2788, ANSI Z39.19.

<sup>4</sup> Теоретически всех, за исключением использующих иероглифическое письмо.

<sup>5</sup> Поддерживаются все версии Oracle, начиная с 9.2.0.3, содержащие новую реализацию Oracle Text.

## Руководство по установке CTX\_API

Установка API может выполняться как с клиента (с использованием Oracle Client/SQL\*Net > 8.1.7), так и с сервера.

**Прerequisites к установке** является наличие установленного Oracle Server (Standard или Enterprise Edition<sup>6</sup>) с установленной опцией Oracle Text и настроенный (вручную или посредством netca/netmgr) SQL\*Net.

**\*\*\*** Установка выполняется с правами пользователя SYS.

Для установки необходимо (в зависимости от целевой платформы) запустить вызывной скрипт **inst\_api.bat/inst\_api.sh**.

При выполнении установочного скрипта запрашиваются:

- Имя целевой схемы, в которую будет произведена установка API;
- Пароль пользователя SYS;
- ORACLE\_SID (SERVICE\_NAME) целевой базы данных<sup>7</sup>;

Целевая схема должна быть создана со следующими правами:

```
grant connect, resource to <целевая схема>;
```

После выполнения установочных скриптов целевой схеме будет переданы - право выполнения пакета CTXSYS.CTX\_THES, роль CTXAPP, и права выборки из представлений CTXSYS.CTX\_THESAURI и CTXSYS.CTX\_THES\_PHRASES.

Подпрограммы API выполняются с правами CURRENT\_USER.

Целевая схема должна оставаться незаблокированной и в целях безопасности для нее должен быть установлен качественный пароль.

Проверка правильности установки API заключается в выполнении следующих команд:

```
SQL> connect <имя целевой схемы>/<пароль>@<ORACLE_SID>  
SQL> select ctx_api.version from dual;
```

Запрос должен вернуть номер версии API.

**Замечание:** Функции version и search\_string\_parser<sup>8</sup> функциональны и без загруженных тезаурусов. Для успешного выполнения остальных функций API требуется наличие хотя бы одного загруженного в базу данных иерархического тезауруса<sup>9</sup>. При отсутствии таковых генерируются исключения **ORA-20154**: No thesaurus found и **ORA-20150**: Oracle Text error. Possible specified thesaurus not loaded.

<sup>6</sup> Теоретически возможна поддержка и Oracle Express Edition (XE), однако использование CTX\_API на данной версии нецелесообразно, за исключением разработки приложений. Тем не менее, ввиду серьезной функциональной ограниченности Oracle XE подобная установка не рекомендуется.

<sup>7</sup> Перед установкой желательно убедиться при помощи утилиты TNSPING в точности имени сервиса/имени экземпляра БД и в его доступности.

<sup>8</sup> Только в режиме KEYWORD.

<sup>9</sup> **Иерархический тезаурус русского языка** является коммерческим продуктом.

**Список процедур и функций пакета CTX\_API и их краткое описание**

Подпрограммы пакета CTX\_API логически разделены на три функциональные группы: API version, Thesaurus CTX API, Thesaurus content API.

Функциональные описания подпрограмм пакета приведены в таблице 1.

Таблица 1

Подпрограмма пакета	Функциональное описание
function <b>version</b>	Возвращает номер версии интерфейса. Используется для проверки функциональной совместимости.
function <b>phrase_exists</b>	Возвращает true, если заданный термин существует в заданном тезаурусе. Используется для проверки существования термина в тезаурусе.
function <b>phrase_relation_exists</b>	Функция возвращает уровень NT/VT, на котором относительно заданного термина находится более чем c_nt_terms одноуровневых терминов NT в дереве иерархии тезауруса (c_nt_terms по умолчанию равно 5).
function <b>search_expansion_term</b>	Функция возвращает родительский термин VT субкатегории, для которой относительно заданного термина находится более чем c_nt_terms одноуровневых терминов NT в дереве иерархии тезауруса (c_nt_terms по умолчанию равно 5).
function <b>has_homographs</b>	Функция проверяет существование гомографов для заданного термина. Возвращает true если существует хотя бы один гомограф, и false в противном случае.
procedure <b>get_qualifiers</b>	Процедура возвращает квалификатор для заданного термина, если квалификатор существует. Предназначена для определения родительской субкатегории для гомографов в расширительных запросах
function <b>get_note</b>	Функция возвращает Scope Note (SN) – комментарий для заданного термина в заданном тезаурусе. Если SN не существует, возвращается пустая строка. Если термин не существует или тезаурус не загружен, возвращается исключение ORA-20151.
function <b>get_bt</b>	Функция возвращает субкатегорию для заданного термина (единственный термин). Если термин отсутствует в тезаурусе, функция возвращает только его. Если термин имеет гомографы, но ни один из них не был задан при вызове, возвращается исключение ORA-20152. Если термин задан с квалификатором, возвращается субкатегория VT заданного уровня.
procedure <b>get_bt</b>	Процедура возвращает <b>ВСЕ</b> субкатегории VT для заданного термина (поддерева VT). Если термин отсутствует в тезаурусе, процедура возвращает только его. Если термин имеет гомографы, но при вызове процедуры не задан квалификатор, возвращаются все субкатегории VT (поддерева VT) для каждого гомографа, причем каждое поддерево начинается с заданного термина с квалификатором данной ветви. Поддерева выводятся в обратной последовательности (то есть на вершине находится заданный термин, последующие субкатегории VT идут в обратном порядке до заданного при вызове процедуры уровня или до вершины дерева иерархии, если термин

	находится на уровне, меньшем чем заданная величина уровня иерархии). Если термин имеет гомографы и при вызове процедуры задан квалификатор, процедура возвращает только поддерево иерархии ВТ заданного термина.
procedure <b>get_nt</b>	Процедура возвращает NT-термины для заданного термина. Если термин отсутствует в тезаурусе, возвращается исключение ORA-20151. Если термин имеет гомографы, но квалификатор не задан, возвращается исключение ORA-20152. Если термин имеет гомографы и задан квалификатор, возвращается поддерево NT-терминов заданного уровня расширения. Если термин находится на самом нижнем уровне иерархии (не имеет NT-терминов), возвращается только заданный термин.
procedure <b>get_ntp</b>	Процедура возвращает NTP-термины для заданного термина. Если термин отсутствует в тезаурусе, возвращается исключение ORA-20151. Если термин имеет гомографы, но квалификатор не задан, возвращается исключение ORA-20152. Если термин имеет гомографы и задан квалификатор, возвращается поддерево NTP-терминов заданного уровня расширения. Если термин находится на самом нижнем уровне иерархии (не имеет NTP-терминов), возвращается только заданный термин.
procedure <b>get_rt</b>	Процедура возвращает ассоциативные термины RT для заданного термина. Если термин отсутствует в тезаурусе, возвращается исключение ORA-20151. Если термин имеет гомографы, но квалификатор не задан, возвращается исключение ORA-20152. Если термин имеет гомографы и задан квалификатор, возвращается все его ассоциативные отношения RT. Если термин не имеет RT-терминов, возвращается только заданный термин.
procedure <b>get_syn</b>	Процедура возвращает синонимы SYN для заданного термина. Если термин отсутствует в тезаурусе, возвращается исключение ORA-20151. Если термин имеет гомографы, но квалификатор не задан, возвращается исключение ORA-20152. Если термин имеет гомографы и задан квалификатор, возвращается все его синонимы. Если термин не имеет синонимов, возвращается только заданный термин.
function <b>search_string_parser</b>	Функция универсального высокоскоростного парсера. Поддерживает множественные тезаурусы, в режиме KEYWORD (по умолчанию) работает без тезаурусов, поддерживает основные отношения стандартов ISO-2788 и ANSI Z39.19. Также поддерживает все логические операнды, словосочетания и квалификаторы гомографов.
function <b>term_counter</b>	Функция возвращает количество уникальных терминов для заданного тезауруса. Если заданный тезаурус не загружен, возвращается исключение ORA-20150.
procedure <b>thes_loaded</b>	Процедура возвращает список загруженных тезаурусов. Если не загружен ни один тезаурус, возвращается исключение ORA-20154.

## Использование подпрограмм пакета CTX\_API

API version

function version return varchar2 deterministic;

- Возвращает номер версии интерфейса. Используется для проверки функциональной совместимости.

Функция может быть вызвана как из PL/SQL, так и из SQL.

Пример:

```
SQL> col "CTX API Version" format a20
SQL> select ctx_api.version as "CTX API Version" from dual;
```

```
CTX API Version
-----
1.0.0.5
```

WSDL-интерфейс возвращает значение данной функции в формате SOAP-сообщения:

```
<?xml version="1.0" encoding="UTF-8" ?>
= <SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema">
= <SOAP-ENV:Body>
  = <ns1:versionResponse xmlns:ns1="http://yvoinov.com/ctx_api.wsdl" SOAP-
    ENV:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/">
    <return xsi:type="xsd:string">1.0.0.5</return>
  </ns1:versionResponse>
</SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

Thesaurus CTX API

Константы, переменные и типы пакета:

```
-- Package constants
c_query_op_about constant varchar2(5) := 'about'; -- ABOUT query option
c_query_op_bt constant varchar2(2) := 'bt'; -- BT query option
c_query_op_nt constant varchar2(2) := 'nt'; -- NT query option
c_query_op_rt constant varchar2(2) := 'rt'; -- RT query option
c_query_op_syn constant varchar2(3) := 'syn'; -- SYN query option

c_refine_on constant number(1) := 1; -- Context refiner ON
c_refine_off constant number(1) := 0; -- Context refiner OFF

c_exp_detail_on constant number(1) := 1; -- Context expansion ON
c_exp_detail_off constant number(1) := 0; -- Context expansion OFF

c_nt_terms constant number(2) := 5; -- Expansion level stop quantity.
-- Stop expansion level if NT's
-- in subtree more than that constant.

-- CTX API types
type term_tab is table of varchar2(256) index by binary_integer;

-- Thesaurus content API
type thes_tab is table of varchar2(30) index by binary_integer;
```

Константа `c_nt_terms` определяет количество терминов NT для субкатегории, на которой должно прекратиться выполнение расширительных функций **search\_expansion\_level** и **search\_expansion\_term**. Данную константу также использует парсер **search\_string\_parser** в режиме `p_exp_detail_on` in number default `ctx_api.c_exp_detail_on`.

```
function phrase_exists (p_phrase in varchar2,
                       p_thes_name in varchar2
                       default 'default') return boolean
deterministic;
```

- Функция проверяет, существует ли в тезаурусе заданный термин.  
Аргументы:  
**p\_phrase** используется для задания термина, может содержать квалификатор.  
**p\_thes\_name** – тезаурус, для которого выполняется функция. По умолчанию – 'default'.

Функция может быть вызвана только из PL/SQL.

Пример:

```
SQL> declare
2   v_exists boolean;
3   begin
4   v_exists := ctx_api.phrase_exists('яблоко');
5   if v_exists then
6   dbms_output.put_line('Phrase exists');
7   else
8   dbms_output.put_line('Phrase NOT exists');
9   end if;
10  end;
11  /
```

Phrase exists

PL/SQL procedure successfully completed.

WSDL-вызовы требуют указания всех аргументов явно:

## phraseExists

### Test

To test the operation using the HTTP GET protocol, click the 'Invoke' button.

Parameter	Type	Value
param0	string	<input type="text" value="яблоко"/>
param1	string	<input type="text" value="default"/>

Результат вызова функции через WSDL:

```
<?xml version="1.0" encoding="UTF-8" ?>
= <SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema">
= <SOAP-ENV:Body>
  = <ns1:phraseExistsResponse xmlns:ns1="http://yvoinov.com/ctx_api.wsdl" SOAP-
    ENV:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/">
    <return xsi:type="xsd:boolean">true</return>
  </ns1:phraseExistsResponse>
</SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

```
function phrase_relation_exists (p_phrase in varchar2,
                                p_relation in varchar2
                                default 'bt,btp,nt,ntp,rt,syn',
                                p_thes_name in varchar2
                                default 'default') return boolean
deterministic;
```

- Функция проверяет, существует ли в тезаурусе заданное отношение (отношения) для заданного термина.  
Аргументы:  
**p\_phrase** используется для задания термина, может содержать квалификатор.  
**p\_relation** представляет собой строковый список отношений, на наличие которых проверяется термин. По умолчанию равен 'bt,btp,nt,ntp,rt,syn'<sup>10</sup>.  
**p\_thes\_name** - тезаурус, для которого выполняется функция. По умолчанию - 'default'.

Функция может быть вызвана только из PL/SQL.

Пример:

```
SQL> declare
2   result boolean;
3   begin
4       dbms_output.put_line('Проверка существования отношений BT/NT/RT/SYN для
заданной фразы');
5
6       result := ctx_api.phrase_relation_exists('котопес');
7       if (result) then dbms_output.put_line('Отношения есть');
8           else dbms_output.put_line('Отношений НЕТ');
9       end if;
10  end;
11  /
```

Проверка существования отношений BT/NT/RT/SYN для заданной фразы  
Отношения есть

PL/SQL procedure successfully completed.

WSDL-вызовы требуют указания всех аргументов явно:

Click [here](#) for a complete list of operations.

## phraseRelationExists

### Test

To test the operation using the HTTP GET protocol, click the 'Invoke' button.

Parameter	Type	Value
param0	string	<input type="text" value="котопес"/>
param1	string	<input type="text" value="bt,btp,nt,ntp,rt,syn"/>
param2	string	<input type="text" value="default"/>

Результат вызова функции через WSDL:

```
<?xml version="1.0" encoding="UTF-8" ?>
= <SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema">
```

<sup>10</sup> Значение по умолчанию - список основных отношений иерархического тезауруса русского языка.



```
= <SOAP-ENV:Body>
= <ns1:phraseRelationExistsResponse xmlns:ns1="http://yvoinov.com/ctx_api.wsdl"
  SOAP-ENV:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/">
  <return xsi:type="xsd:boolean">true</return>
</ns1:phraseRelationExistsResponse>
</SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

```
function search_expansion_level (p_phrase in varchar2,
                                p_thes_name in varchar2
                                default 'default') return number
deterministic;
```

- Функция возвращает уровень NT/BT, на котором относительно заданного термина находится более чем с\_nt\_terms одноуровневых терминов NT в дереве иерархии тезауруса (с\_nt\_terms по умолчанию равно 5).

Аргументы:

**p\_phrase** используется для задания термина, может содержать квалификатор.

**p\_thes\_name** - тезаурус, для которого выполняется функция. По умолчанию - 'default'.

Функция может быть вызвана из PL/SQL и SQL.

Пример:

```
SQL> set serveroutput on
SQL>
SQL> declare
  2   result number;
  3   begin
  4     result := ctx_api.search_expansion_level('динозавр');
  5     dbms_output.put_line(result);
  6   end;
  7   /
2
```

PL/SQL procedure successfully completed.

При вызове через WSDL указываются все аргументы:

Click [here](#) for a complete list of operations.

## searchExpansionLevel

### Test

To test the operation using the HTTP GET protocol, click the 'Invoke' button.

Parameter	Type	Value
param0	string	<input type="text" value="динозавр"/>
param1	string	<input type="text" value="default"/>

Результат вызова функции через WSDL:

```
<?xml version="1.0" encoding="UTF-8" ?>
= <SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema">
= <SOAP-ENV:Body>
  = <ns1:searchExpansionLevelResponse xmlns:ns1="http://yvoinov.com/ctx_api.wsdl"
    SOAP-ENV:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/">
    <return xsi:type="xsd:decimal">2</return>
  </ns1:searchExpansionLevelResponse>
</SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

```
function search_expansion_term (p_phrase in varchar2,
                                p_thes_name in varchar2
                                default 'default') return varchar2
deterministic;
```

- Функция возвращает родительский термин ВТ субкатегории, для которой относительно заданного термина находится более чем с\_nt\_terms одноуровневых терминов NT в дереве иерарархии тезауруса (с\_nt\_terms по умолчанию равно 5).

Аргументы:

**p\_phrase** используется для задания термина, может содержать квалификатор.

**p\_thes\_name** - тезаурус, для которого выполняется функция. По умолчанию - 'default'.

Функция может быть вызвана из PL/SQL и SQL.

Пример:

```
SQL> declare
2   result varchar2(255);
3   begin
4   result := ctx_api.search_expansion_term('динозавр');
5   dbms_output.put_line(result);
6   end;
7   /
```

ПАЛЕОЗООЛОГИЯ

PL/SQL procedure successfully completed.

При вызове через WSDL указываются все аргументы:

Click [here](#) for a complete list of operations.

## searchExpansionTerm

### Test

To test the operation using the HTTP GET protocol, click the 'Invoke' button.

Parameter	Type	Value
param0	string	<input type="text" value="динозавр"/>
param1	string	<input type="text" value="default"/>

Результат вызова функции через WSDL:

```
<?xml version="1.0" encoding="UTF-8" ?>
= <SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema">
= <SOAP-ENV:Body>
  = <ns1:searchExpansionTermResponse xmlns:ns1="http://yvoinov.com/ctx_api.wsdl"
    SOAP-ENV:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/">
    <return xsi:type="xsd:string">ПАЛЕОЗООЛОГИЯ</return>
  </ns1:searchExpansionTermResponse>
</SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

```
function has_homographs (p_phrase in varchar2,
                        p_thes_name in varchar2
                        default 'default') return boolean
deterministic;
```

- Функция проверяет существование гомографов для заданного термина. Возвращает true если существует хотя бы один гомограф, и false в противном случае.

Аргументы:

**p\_phrase** используется для задания термина, может содержать квалификатор.

**P\_thes\_name** - тезаурус, для которого выполняется функция. По умолчанию - 'default'.

Функция может быть вызвана только из PL/SQL.

Пример:

```
SQL> set serveroutput on
SQL>
SQL> declare
2   v_hom boolean;
3   begin
4   v_hom := ctx_api.has_homographs('якорь');
5   if v_hom then dbms_output.put_line('Гомографы есть');
6               else dbms_output.put_line('Гомографов НЕТ');
7   end if;
8   end;
9   /
```

Гомографы есть

PL/SQL procedure successfully completed.

```
SQL> declare
2   v_hom boolean;
3   begin
4   v_hom := ctx_api.has_homographs('якорь (флот)');
5   if v_hom then dbms_output.put_line('Гомографы есть');
6               else dbms_output.put_line('Гомографов НЕТ');
7   end if;
8   end;
9   /
```

Гомографы есть

PL/SQL procedure successfully completed.

```
SQL> declare
2   v_hom boolean;
3   begin
4   v_hom := ctx_api.has_homographs('змеи');
5   if v_hom then dbms_output.put_line('Гомографы есть');
6               else dbms_output.put_line('Гомографов НЕТ');
7   end if;
8   end;
9   /
```

Гомографов НЕТ

PL/SQL procedure successfully completed.

При вызове через WSDL указываются все аргументы<sup>11</sup>:

<sup>11</sup> При вызове в рамках B2B сервиса аргументы передаются через URL: <http://<hostname>/wsdl/ctxapi?invoke=hasHomographs&param0=крыло&param1=default>

Click [here](#) for a complete list of operations.

## hasHomographs

### Test

To test the operation using the HTTP GET protocol, click the 'Invoke' button.

Parameter	Type	Value
param0	string	<input type="text" value="крыло"/>
param1	string	<input type="text" value="default"/>

Результат вызова функции через WSDL:

```
<?xml version="1.0" encoding="UTF-8" ?>
= <SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema">
= <SOAP-ENV:Body>
  = <ns1:hasHomographsResponse xmlns:ns1="http://yvoinov.com/ctx_api.wsdl" SOAP-
    ENV:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/">
    <return xsi:type="xsd:boolean">true</return>
  </ns1:hasHomographsResponse>
</SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

```
procedure get_qualifiers ( p_qualifiers out qual_tab,
                          p_phrase in varchar2,
                          p_thes_name in varchar2 default 'default');
```

- Процедура возвращает квалификатор для заданного термина, если квалификатор существует. Предназначена для определения родительской субкатегории для томографов в расширительных запросах.

Аргументы:

**p\_qualifiers** содержит массив возвращаемых значений (терминов) типа ctx\_api.term\_tab.

**p\_phrase** используется для задания термина, может содержать квалификатор.

**p\_thes\_name** - тезаурус, для которого выполняется функция. По умолчанию - 'default'.

Процедура может быть вызвана только из PL/SQL.

Пример:

```
SQL> set serveroutput on
SQL>
SQL> declare
  2   v_hom ctx_api.term_tab;
  3   i pls_integer;
  4   begin
  5     ctx_api.get_qualifiers(v_hom,'якорь','default');
  6     for i in v_hom.first..v_hom.last loop
  7       dbms_output.put_line(v_hom(i));
  8     end loop;
  9   end;
 10  /
```

ФЛОТ

ЭЛЕКТРОТЕХНИКА

PL/SQL procedure successfully completed.

```
SQL> declare
  2   v_hom ctx_api.term_tab;
  3   i pls_integer;
  4   begin
  5     ctx_api.get_qualifiers(v_hom,'варкалось','default');
  6     for i in v_hom.first..v_hom.last loop
  7       dbms_output.put_line(v_hom(i));
  8     end loop;
  9   end;
 10  /
```

declare

\*

ERROR at line 1:

ORA-20151: Phrase "ВАРКАЛОСЬ" not exist or specified thesaurus "DEFAULT" not loaded.

ORA-06512: at "SCOTT.CTX\_API", line 166

ORA-06512: at line 5

При вызове через WSDL указываются все аргументы:

Click [here](#) for a complete list of operations.

## getQualifiers

### Test

To test the operation using the HTTP GET protocol, click the 'Invoke' button.

Parameter	Type	Value
param0	string	<input type="text" value="якорь"/>
param1	string	<input type="text" value="default"/>

Click [here](#) for a complete list of operations.

## getQualifiers

### Test

To test the operation using the HTTP GET protocol, click the 'Invoke' button.

Parameter	Type	Value
param0	string	<input type="text" value="варкалось"/>
param1	string	<input type="text" value="default"/>

Результаты вызовов процедуры через WSDL:

```
<?xml version="1.0" encoding="UTF-8" ?>
= <SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema">
= <SOAP-ENV:Body>
  = <ns1:getQualifiersResponse xmlns:ns1="http://yvoinov.com/ctx_api.wsdl" SOAP-
    ENV:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/">
    = <return xmlns:ns2="http://yvoinov.com/ctx_api.xsd"
      xsi:type="ns2:yvoinov_com_ctx_apiUser_getQualifiers_Out">
      = <pqualifiersOut xsi:type="ns2:yvoinov_com_CtxApiTermTab">
        = <array xmlns:ns3="http://schemas.xmlsoap.org/soap/encoding/"
          xsi:type="ns3:Array" ns3:arrayType="xsd:string[2]">
          <item xsi:type="xsd:string">ФЛОТ</item>
          <item xsi:type="xsd:string">ЭЛЕКТРОТЕХНИКА</item>
        </array>
      </pqualifiersOut>
    </return>
  </ns1:getQualifiersResponse>
</SOAP-ENV:Body>
</SOAP-ENV:Envelope>

<?xml version="1.0" encoding="UTF-8" ?>
= <SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema">
= <SOAP-ENV:Body>
  = <SOAP-ENV:Fault>
    <faultcode>SOAP-ENV:Server.Exception:</faultcode>
    <faultstring>java.sql.SQLException: java.sql.SQLException: ORA-20151: Phrase
      "ВАРКАЛОСЬ" not exist or specified thesaurus "DEFAULT" not loaded.
      ORA-06512: at "WSDL_OWN.CTX_API", line 166 ORA-06512: at
```

```
"WSDL_OWN.JPUB_PLSQL_WRAPPER", line 55 ORA-06512: at line
1</faultstring>
<faultactor>/wsdl/ctxapi</faultactor>
</SOAP-ENV:Fault>
</SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```



```
function get_note (p_phrase in varchar2,
                  p_thes_name in varchar2
                  default 'default') return varchar2
deterministic;
```

- Функция возвращает Scope Note (SN) для заданного термина, если SN определен (существует). Предназначена для извлечения комментария (SN), если он задан для термина. Если термин не существует или тезаурус не загружен, возвращается исключение ORA-20151.

Аргументы:

**p\_phrase** используется для задания термина, может содержать квалификатор.

**P\_thes\_name** - тезаурус, для которого выполняется функция. По умолчанию - 'default'.

Процедура может быть вызвана только из PL/SQL.

При вызове через WSDL указываются все аргументы:

## getNote

### Test

To test the operation using the HTTP GET protocol, click the 'Invoke' button.

Parameter	Type	Value
param0	string	<input type="text" value="Эффект хотторна"/>
param1	string	<input type="text" value="default"/>

Результаты вызовов процедуры через WSDL:

```
<?xml version="1.0" encoding="UTF-8" ?>
= <SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  = <SOAP-ENV:Body>
    = <ns1:getNoteResponse xmlns:ns1="http://yvoinov.com/ctx_api.wsdl" SOAP-
      ENV:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/">
      <return xsi:type="xsd:string">Эффект хотторна - Новизна, интерес к
        эксперименту или повышенное внимание к исследуемому вопросы
        приводит к искаженному или слишком благоприятному исходу
        эксперимента.</return>
    </ns1:getNoteResponse>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

```
function get_bt (p_phrase in varchar2,
                p_level in number default 1,
                p_thes_name in varchar2 default 'default')
return varchar2 deterministic;
```

- Функция возвращает субкатегорию для заданного термина (единственный термин). Если термин отсутствует в тезаурусе, функция возвращает только его. Если термин имеет омографы, но ни один из них не был задан при вызове, возвращается исключение ORA-20152. Если термин задан с квалификатором, возвращается субкатегория ВТ заданного уровня.

Аргументы:

**p\_phrase** используется для задания термина, может содержать квалификатор.

**p\_level** уровень иерархии в отношениях ВТ. Значение по умолчанию 1.

**p\_thes\_name** - тезаурус, для которого выполняется функция. По умолчанию - 'default'.

Функция может быть вызвана из PL/SQL и SQL.

Примеры:

```
SQL> select ctx_api.get_bt('кот') as "Категория" from dual;
```

Категория

-----  
КОШКА

```
SQL> select ctx_api.get_bt('крыло') as "Категория" from dual;
select ctx_api.get_bt('крыло') as "Категория" from dual
      *
```

```
ERROR at line 1:
ORA-20152: Phrase "КРЫЛО" has homographs.
ORA-06512: at "SCOTT.CTX_API", line 196
ORA-06512: at line 1
```

```
SQL> select ctx_api.get_bt('крыло (авиация)') as "Категория" from dual;
```

Категория

-----  
ПЛАНЕР (ЧАСТИ ЛЕТАТЕЛЬНЫХ АППАРАТОВ)

При вызове через WSDL указываются все аргументы.

**\*\*\* ВНИМАНИЕ!** Следует иметь в виду, что в WSDL-интерфейсе данная функция имеет имя **getBtSSBS** !

Click [here](#) for a complete list of operations.

## getBtSSBS

### Test

To test the operation using the HTTP GET protocol, click the 'Invoke' button.

Parameter	Type	Value
param0	string	<input type="text" value="кот"/>
param1	decimal	<input type="text" value="5"/>
param2	string	<input type="text" value="default"/>

Результат вызова функции через WSDL:

```
<?xml version="1.0" encoding="UTF-8" ?>
= <SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema">
= <SOAP-ENV:Body>
  = <ns1:getBtSSBSResponse xmlns:ns1="http://yvoinov.com/ctx_api.wsdl" SOAP-
    ENV:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/">
    <return xsi:type="xsd:string">БИОЛОГИЯ</return>
  </ns1:getBtSSBSResponse>
</SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

Как следует из данного примера, функция возвращает 5 уровней вверх по иерархии ВТ тезауруса вплоть до субкатегории **БИОЛОГИЯ** и возвращает непосредственно субкатегорию 5 уровня для заданного термина.

```
procedure get_bt (p_bt out ctx_api.term_tab,
                 p_phrase in varchar2,
                 p_level in number default 1,
                 p_thes_name in varchar2 default 'default');
```

- Процедура возвращает **ВСЕ** субкатегории ВТ для заданного термина (поддеревья ВТ). Если термин отсутствует в тезаурусе, процедура возвращает только его. Если термин имеет гомографы, но при вызове процедуры не задан квалификатор, возвращаются все субкатегории ВТ (поддеревья ВТ) для каждого гомографа, причем каждое поддерево начинается с заданного термина с квалификатором данной ветви. Поддеревья выводятся в обратной последовательности (то есть на вершине находится заданный термин, последующие субкатегории ВТ идут в обратном порядке до заданного при вызове процедуры уровня или до вершины дерева иерархии, если термин находится на уровне, меньшем чем заданная величина уровня иерархии). Если термин имеет гомографы и при вызове процедуры задан квалификатор, процедура возвращает только поддерево иерархии ВТ заданного термина.

Аргументы:

**p\_bt** содержит массив возвращаемых значений (терминов) типа ctx\_api.term\_tab.  
**p\_phrase** используется для задания термина, может содержать квалификатор.  
**p\_level** уровень иерархии в отношениях ВТ. Значение по умолчанию 1.  
**p\_thes\_name** - тезаурус, для которого выполняется функция. По умолчанию - 'default'.

Процедура может быть вызвана только из PL/SQL.

Примеры:

```
SQL> declare
2   xtab ctx_api.term_tab;
3   begin
4   -- Термин с гомографами - выводятся оба субдерева ВТ одно за другим
5   ctx_api.get_bt(xtab, 'алмаз', 5, 'default');
6   for i in 1..xtab.count loop
7   dbms_output.put_line(xtab(i));
8   end loop;
9   end;
10  /
```

```
АЛМАЗ (ГЕОЛОГИЯ)
МИНЕРАЛЫ
МИНЕРАЛОГИЯ
ГЕОЛОГИЯ
ЕСТЕСТВОЗНАНИЕ
НАУКА
АЛМАЗ (МЕТАЛЛООБРАБОТКА)
СТАНКИ (МЕТАЛЛООБРАБОТКА)
МЕТАЛЛООБРАБОТКА
ПРОМЫШЛЕННОСТЬ И ПРОИЗВОДСТВО
```

PL/SQL procedure successfully completed.

```
SQL> declare
2   xtab ctx_api.term_tab;
3   begin
4   -- Термин с гомографами - квалифицированная подветвь одного из поддеревьев
5   ctx_api.get_bt(xtab, 'алмаз (геология)', 5, 'default');
6   for i in 1..xtab.count loop
7   dbms_output.put_line(xtab(i));
8   end loop;
9   end;
10  /
```

```
МИНЕРАЛЫ
МИНЕРАЛОГИЯ
ГЕОЛОГИЯ
ЕСТЕСТВОЗНАНИЕ
НАУКА
```

```
SQL> declare
2   xtab ctx_api.term_tab;
```

```

3  begin
4    -- Термин с гомографами - вторая квалифицированная подветвь
5    ctx_api.get_bt(xtab, 'алмаз (металлообработка)', 5, 'default');
6    for i in 1..xtab.count loop
7      dbms_output.put_line(xtab(i));
8    end loop;
9  end;
10 /
СТАНКИ (МЕТАЛЛООБРАБОТКА)
МЕТАЛЛООБРАБОТКА
ПРОМЫШЛЕННОСТЬ И ПРОИЗВОДСТВО

PL/SQL procedure successfully completed.

```

При вызове через WSDL указываются все аргументы:

Click [here](#) for a complete list of operations.

## getBt

### Test

To test the operation using the HTTP GET protocol, click the 'Invoke' button.

Parameter	Type	Value
param0	string	алмаз
param1	decimal	5
param2	string	default

Результат вызова процедуры через WSDL:

```

<?xml version="1.0" encoding="UTF-8" ?>
= <SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  = <SOAP-ENV:Body>
    = <ns1:getBtResponse xmlns:ns1="http://yvoinov.com/ctx_api.wsdl" SOAP-
      ENV:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/">
      = <return xmlns:ns2="http://yvoinov.com/ctx_api.xsd"
        xsi:type="ns2:yvoinov_com_ctx_apiUser_getBt_Out">
        = <pbtOut xsi:type="ns2:yvoinov_com_CtxApiTermTab">
          = <array xmlns:ns3="http://schemas.xmlsoap.org/soap/encoding/"
            xsi:type="ns3:Array" ns3:arrayType="xsd:string[10]">
            <item xsi:type="xsd:string">АЛМАЗ (ГЕОЛОГИЯ)</item>
            <item xsi:type="xsd:string">МИНЕРАЛЫ</item>
            <item xsi:type="xsd:string">МИНЕРАЛОГИЯ</item>
            <item xsi:type="xsd:string">ГЕОЛОГИЯ</item>
            <item xsi:type="xsd:string">ЕСТЕСТВОЗНАНИЕ</item>
            <item xsi:type="xsd:string">НАУКА</item>
            <item xsi:type="xsd:string">АЛМАЗ (МЕТАЛЛООБРАБОТКА)</item>
            <item xsi:type="xsd:string">СТАНКИ (МЕТАЛЛООБРАБОТКА)</item>
            <item xsi:type="xsd:string">МЕТАЛЛООБРАБОТКА</item>
            <item xsi:type="xsd:string">ПРОМЫШЛЕННОСТЬ И
              ПРОИЗВОДСТВО</item>
          </array>
        </pbtOut>
      </return>
    </ns1:getBtResponse>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>

```

Как видно из данного примера, процедура возвращает оба квалифицированных иерархических поддеревя отношений ВТ для термина АЛМАЗ на максимальное количество уровней вверх по иерархии тезауруса равное 5.

```
procedure get_nt (p_nt out ctx_api.term_tab,
                 p_phrase in varchar2,
                 p_level in number default 1,
                 p_thes_name in varchar2 default 'default');
```

- Процедура возвращает NT-термины для заданного термина. Если термин отсутствует в тезаурусе, возвращается исключение ORA-20151. Если термин имеет гомографы, но квалификатор не задан, возвращается исключение ORA-20152. Если термин имеет гомографы и задан квалификатор, возвращается поддерево NT-терминов заданного уровня расширения. Если термин находится на самом нижнем уровне иерархии (не имеет NT-терминов), возвращается только заданный термин.

Аргументы:

**p\_nt** содержит массив возвращаемых значений (терминов) типа ctx\_api.term\_tab.

**p\_phrase** используется для задания термина, может содержать квалификатор.

**p\_level** уровень иерархии в отношениях ВТ. Значение по умолчанию 1.

**p\_thes\_name** - тезаурус, для которого выполняется функция. По умолчанию - 'default'.

Процедура может быть вызвана только из PL/SQL.

Пример:

```
SQL> declare
2   xtab ctx_api.term_tab;
3   i number;
4   begin
5   ctx_api.get_nt(xtab,'планер (части летательных аппаратов)',1,'default');
6   if xtab.count > 0 then dbms_output.put_line('Has '||xtab.count||'NT's');end
if;
7   for i in 1..xtab.last loop
8       dbms_output.put_line('NT('||i||')='||xtab(i));
9   end loop;
10  end;
11  /
Has 6 NT's
NT(1)=КРЫЛО (АВИАЦИЯ)
NT(2)=ФЮЗЕЛЯЖ
NT(3)=ЦЕНТРОПЛАН
NT(4)=МИДЕЛЬ
NT(5)=ЛОНЖЕРОН (АВИАЦИЯ)
NT(6)=ХВОСТОВОЕ ОПЕРЕНИЕ
```

При вызове через WSDL указываются все аргументы:

Click [here](#) for a complete list of operations.

## getNt

## Test

To test the operation using the HTTP GET protocol, click the 'Invoke' button.

Parameter	Type	Value
param0	string	<input type="text" value="планер (части летательных аппаратов)"/>
param1	decimal	<input type="text" value="1"/>
param2	string	<input type="text" value="default"/>

Результат вызова процедуры через WSDL:

```
<?xml version="1.0" encoding="UTF-8" ?>
```

```
= <SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema">
= <SOAP-ENV:Body>
  = <ns1:getNtResponse xmlns:ns1="http://yvoinov.com/ctx_api.wsdl" SOAP-
    ENV:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/">
    = <return xmlns:ns2="http://yvoinov.com/ctx_api.xsd"
      xsi:type="ns2:yvoinov_com_ctx_apiUser_getNt_Out">
      = <pntOut xsi:type="ns2:yvoinov_com_CtxApiTermTab">
        = <array xmlns:ns3="http://schemas.xmlsoap.org/soap/encoding/"
          xsi:type="ns3:Array" ns3:arrayType="xsd:string[6]">
          <item xsi:type="xsd:string">КРЫЛО (АВИАЦИЯ)</item>
          <item xsi:type="xsd:string">ФЮЗЕЛЯЖ</item>
          <item xsi:type="xsd:string">ЦЕНТРОПЛАН</item>
          <item xsi:type="xsd:string">МИДЕЛЬ</item>
          <item xsi:type="xsd:string">ЛОНЖЕРОН (АВИАЦИЯ)</item>
          <item xsi:type="xsd:string">ХВОСТОВОЕ ОПЕРЕНИЕ</item>
        </array>
      </pntOut>
    </return>
  </ns1:getNtResponse>
</SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

```
procedure get_ntp (p_ntp out ctx_api.term_tab,
                  p_phrase in varchar2,
                  p_level in number default 1,
                  p_thes_name in varchar2 default 'default');
```

- Процедура возвращает NTP-термины для заданного термина. Если термин отсутствует в тезаурусе, возвращается исключение ORA-20151. Если термин имеет гомографы, но квалификатор не задан, возвращается исключение ORA-20152. Если термин имеет гомографы и задан квалификатор, возвращается поддерево NTP-терминов заданного уровня расширения. Если термин находится на самом нижнем уровне иерархии (не имеет NTP-терминов), возвращается только заданный термин.

Аргументы:

**p\_ntp** содержит массив возвращаемых значений (терминов) типа tx\_api.term\_tab.

**p\_phrase** используется для задания термина, может содержать квалификатор.

**p\_level** уровень иерархии в отношениях ВТ. Значение по умолчанию 1.

**p\_thes\_name** - тезаурус, для которого выполняется функция. По умолчанию - 'default'.

Процедура может быть вызвана только из PL/SQL.

Пример:

```
1 declare
2     xtab ctx_api.term_tab;
3     i number;
4     begin
5         ctx_api.get_ntp(xtab,'вексель',1,'default'); -- Нет NTP, нижний уровень
6         if xtab.count > 0 then dbms_output.put_line('Has '||xtab.count||' NTP's');
end if;
7         for i in 1..xtab.last loop
8             dbms_output.put_line('NTP('||i||')='||xtab(i));
9         end loop;
10* end;
SQL> /
Has 20 NTP's
NTP(1)=АНТИДАТИРОВАНИЕ
NTP(2)=АНТИДАТИРОВАТЬ
NTP(3)=ВЕКСЕЛЬНЫЕ ВОЗРАЖЕНИЯ
NTP(4)=ДОЛГОВОЕ ОБЯЗАТЕЛЬСТВО
NTP(5)=ОБОРОТНЫЕ ДОКУМЕНТЫ
NTP(6)=ЯРМАРОЧНЫЙ ВЕКСЕЛЬ
NTP(7)=ДАМНО
NTP(8)=ПЕРЕУЧЕТНАЯ ОПЕРАЦИЯ
NTP(9)=ДЕНЕЖНЫЕ СУРРОГАТЫ
NTP(10)=РАМБУРСИРОВАТЬ
NTP(11)=ГРАЦИЯ ВЕСКЕЛЬНАЯ
NTP(12)=ДЕНЕЖНЫЕ ЦЕННЫЕ БУМАГИ
NTP(13)=ДОЛГОВАЯ ЦЕННАЯ БУМАГА
NTP(14)=ПЛАТЕЖНЫЙ ДОКУМЕНТ
NTP(15)=ВСТРЕЧНЫЙ ВЕКСЕЛЬ
NTP(16)=ВЗАИМНЫЙ ВЕКСЕЛЬ
NTP(17)=ДАТА-ВЕКСЕЛЬ
NTP(18)=ДРУЖЕСКИЙ ВЕКСЕЛЬ
NTP(19)=ИНКАССИРОВАНИЕ
NTP(20)=ВЕКСЕЛЬНЫЙ БЛАНК
```

PL/SQL procedure successfully completed.



При вызове через WSDL указываются все аргументы:  
Click [here](#) for a complete list of operations.

## getNtp

### Test

To test the operation using the HTTP GET protocol, click the 'Invoke' button.

Parameter	Type	Value
param0	string	<input type="text" value="вексель"/>
param1	decimal	<input type="text" value="1"/>
param2	string	<input type="text" value="default"/>

Результат вызова процедуры через WSDL:

```
<?xml version="1.0" encoding="UTF-8" ?>
= <SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  = <SOAP-ENV:Body>
    = <ns1:getNtpResponse xmlns:ns1="http://yvoinov.com/ctx_api.wsdl" SOAP-
      ENV:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/">
      = <return xmlns:ns2="http://yvoinov.com/ctx_api.xsd"
        xsi:type="ns2:yvoinov_com_ctx_apiUser_getNtp_Out">
        = <pntpOut xsi:type="ns2:yvoinov_com_CtxApiTermTab">
          = <array xmlns:ns3="http://schemas.xmlsoap.org/soap/encoding/"
            xsi:type="ns3:Array" ns3:arrayType="xsd:string[20]">
            <item xsi:type="xsd:string">АНТИДАТИРОВАНИЕ</item>
            <item xsi:type="xsd:string">АНТИДАТИРОВАТЬ</item>
            <item xsi:type="xsd:string">ВЕКСЕЛЬНЫЕ ВОЗРАЖЕНИЯ</item>
            <item xsi:type="xsd:string">ДОЛГОВОЕ ОБЯЗАТЕЛЬСТВО</item>
            <item xsi:type="xsd:string">ОБОРОТНЫЕ ДОКУМЕНТЫ</item>
            <item xsi:type="xsd:string">ЯРМАРОЧНЫЙ ВЕКСЕЛЬ</item>
            <item xsi:type="xsd:string">ДАМНО</item>
            <item xsi:type="xsd:string">ПЕРЕУЧЕТНАЯ ОПЕРАЦИЯ</item>
            <item xsi:type="xsd:string">ДЕНЕЖНЫЕ СУРРОГАТЫ</item>
            <item xsi:type="xsd:string">РАМБУРСИРОВАТЬ</item>
            <item xsi:type="xsd:string">ГРАЦИЯ ВЕСКЕЛЬНАЯ</item>
            <item xsi:type="xsd:string">ДЕНЕЖНЫЕ ЦЕННЫЕ БУМАГИ</item>
            <item xsi:type="xsd:string">ДОЛГОВАЯ ЦЕННАЯ БУМАГА</item>
            <item xsi:type="xsd:string">ПЛАТЕЖНЫЙ ДОКУМЕНТ</item>
            <item xsi:type="xsd:string">ВСТРЕЧНЫЙ ВЕКСЕЛЬ</item>
            <item xsi:type="xsd:string">ВЗАИМНЫЙ ВЕКСЕЛЬ</item>
            <item xsi:type="xsd:string">ДАТА-ВЕКСЕЛЬ</item>
            <item xsi:type="xsd:string">ДРУЖЕСКИЙ ВЕКСЕЛЬ</item>
            <item xsi:type="xsd:string">ИНКАССИРОВАНИЕ</item>
            <item xsi:type="xsd:string">ВЕКСЕЛЬНЫЙ БЛАНК</item>
          </array>
        </pntpOut>
      </return>
    </ns1:getNtpResponse>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

```
procedure get_rt (p_rt out ctx_api.term_tab,
                 p_phrase in varchar2,
                 p_thes_name in varchar2 default 'default');
```

- Процедура возвращает ассоциативные термины RT для заданного термина. Если термин отсутствует в тезаурусе, возвращается исключение ORA-20151. Если термин имеет омографы, но квалификатор не задан, возвращается исключение ORA-20152. Если термин имеет омографы и задан квалификатор, возвращается все его ассоциативные отношения RT. Если термин не имеет RT-терминов, возвращается только заданный термин.

Аргументы:

**p\_rt** содержит массив возвращаемых значений (терминов) типа tx\_api.term\_tab.

**p\_phrase** используется для задания термина, может содержать квалификатор.

**p\_thes\_name** - тезаурус, для которого выполняется функция. По умолчанию - 'default'.

Процедура может быть вызвана только из PL/SQL.

Пример:

```
SQL> declare
2   xtab ctx_api.term_tab;
3   i number;
4   begin
5   ctx_api.get_rt(xtab,'мифология');
6   if xtab.count > 0 then dbms_output.put_line('Has '||xtab.count||' RT's'); end
if;
7   for i in 1..xtab.last loop
8   dbms_output.put_line('RT('||i||')='||xtab(i));
9   end loop;
10  end;
11  /
Has 4 RT's
RT(1)=ПАНТЕОН
RT(2)=МИФОЛОГЕМА
RT(3)=МИФ
RT(4)=ПЕРВОЧЕЛОВЕК
```

При вызове через WSDL указываются все аргументы:

Click [here](#) for a complete list of operations.

## getRt

## Test

To test the operation using the HTTP GET protocol, click the 'Invoke' button.

Parameter	Type	Value
param0	string	<input type="text" value="мифология"/>
param1	string	<input type="text" value="default"/>

Результат вызова процедуры через WSDL:

```
<?xml version="1.0" encoding="UTF-8" ?>
= <SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema">
= <SOAP-ENV:Body>
= <ns1:getRtResponse xmlns:ns1="http://yvoinov.com/ctx_api.wsdl" SOAP-
  ENV:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/">
= <return xmlns:ns2="http://yvoinov.com/ctx_api.xsd"
  xsi:type="ns2:yvoinov_com_ctx_apiUser_getRt_Out">
```

```

= <prtOut xsi:type="ns2:yvoinov_com_CtxApiTermTab">
  = <array xmlns:ns3="http://schemas.xmlsoap.org/soap/encoding/"
    xsi:type="ns3:Array" ns3:arrayType="xsd:string[4]">
    <item xsi:type="xsd:string">ПАНТЕОН</item>
    <item xsi:type="xsd:string">МИФОЛОГЕМА</item>
    <item xsi:type="xsd:string">МИФ</item>
    <item xsi:type="xsd:string">ПЕРВОЧЕЛОВЕК</item>
  </array>
</prtOut>
</return>
</ns1:getRtResponse>
</SOAP-ENV:Body>
</SOAP-ENV:Envelope>

```

```
procedure get_syn (p_syn out ctx_api.term_tab,
                  p_phrase in varchar2,
                  p_thes_name in varchar2 default 'default');
```

- Процедура возвращает синонимы SYN для заданного термина. Если термин отсутствует в тезаурусе, возвращается исключение ORA-20151. Если термин имеет гомографы, но квалификатор не задан, возвращается исключение ORA-20152. Если термин имеет гомографы и задан квалификатор, возвращается все его синонимы. Если термин не имеет синонимов, возвращается только заданный термин.

Аргументы:

**p\_syn** содержит массив возвращаемых значений (терминов) типа tx\_api.term\_tab.

**p\_phrase** используется для задания термина, может содержать квалификатор.

**p\_thes\_name** - тезаурус, для которого выполняется функция. По умолчанию - 'default'.

Процедура может быть вызвана только из PL/SQL.

Пример:

```
SQL> declare
2   xtab ctx_api.term_tab;
3   i number;
4   begin
5     ctx_api.get_syn(xtab, 'джилинг');
6     if xtab.count > 0 then dbms_output.put_line('Has '||xtab.count||' SYN's'); end
if;
7     for i in 1..xtab.last loop
8       dbms_output.put_line('SYN('||i||')='||xtab(i));
9     end loop;
10  end;
11  /
Has 1 SYN's
SYN(1)=ЛЕВЕРЕДЖ
```

При вызове через WSDL указываются все аргументы:

Click [here](#) for a complete list of operations.

## getSyn

### Test

To test the operation using the HTTP GET protocol, click the 'Invoke' button.

Parameter	Type	Value
param0	string	<input type="text" value="джилинг"/>
param1	string	<input type="text" value="default"/>

Результат вызова процедуры через WSDL:

```
<?xml version="1.0" encoding="UTF-8" ?>
= <SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema">
= <SOAP-ENV:Body>
= <ns1:getSynResponse xmlns:ns1="http://yvoinov.com/ctx_api.wsdl" SOAP-
  ENV:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/">
= <return xmlns:ns2="http://yvoinov.com/ctx_api.xsd"
  xsi:type="ns2:yvoinov_com_ctx_apiUser_getSyn_Out">
= <psynOut xsi:type="ns2:yvoinov_com_CtxApiTermTab">
```

```
= <array xmlns:ns3="http://schemas.xmlsoap.org/soap/encoding/"
  xsi:type="ns3:Array" ns3:arrayType="xsd:string[1]">
  <item xsi:type="xsd:string">ЛЕВЕРЕДЖ</item>
</array>
</psynOut>
</return>
</ns1:getSynResponse>
</SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

```
function search_string_parser (p_search_str in varchar2,
                              p_query_mode in varchar2 default 'keyword',
                              p_logical_op in varchar2 default 'and',
                              p_query_opt in varchar2
                              default ctx_api.c_query_op_about,
                              p_expansion_level in number default 1,
                              p_thes_name in varchar2 default 'default',
                              p_refine_on in number
                              default ctx_api.c_refine_off,
                              p_exp_detail_on in number
                              default ctx_api.c_exp_detail_off)
return varchar2 deterministic;
```

- Функция универсального высокоскоростного парсера. Поддерживает множественные тезаурусы, в режиме KEYWORD (по умолчанию) работает без тезаурусов, поддерживает основные отношения стандартов ISO-2788 и ANSI Z39.19. Также поддерживает все логические операнды, словосочетания и квалификаторы гомографов.

Аргументы:

**p\_search\_string** - поисковая строка, передаваемая в парсер для обработки. Должна быть ограничена длиной 4000 символов (максимальной емкостью внутренних буферов).

**p\_query\_mode** определяет режим работы парсера ('keyword' - поиск по ключевым словам или 'concept' - поиск по тематике на основе тезауруса). Значение по умолчанию - 'keyword' (в данном режиме парсер способен работать без тезауруса). Параметр определяет характер преобразования выходной строки парсера.

**p\_logical\_op** - логический операнд, применяемый к токенам выходной строки. Имеет два значения - 'and' или 'or'. По умолчанию 'and'.

**p\_query\_opt** - режим формирования выходной строки в режиме 'concept'. Задается одно из пяти значений: ctx\_api.c\_query\_op\_about, ctx\_api.c\_query\_op\_bt, ctx\_api.c\_query\_op\_nt, ctx\_api.c\_query\_op\_rt, ctx\_api.c\_query\_op\_syn. Значение по умолчанию - ctx\_api.c\_query\_op\_about. Константы определяют функции Oracle Text, которые будут использоваться при формировании финальной строки, передаваемой для исполнения поисковой системе. Игнорируется в режиме 'keyword'.

**p\_expansion\_level** - уровень иерархии в расширительных функциях bt/nt в режиме 'concept'. Значение по умолчанию - 1. Игнорируется в режиме 'keyword' и в режиме 'concept', если значение p\_query\_opt равно ctx\_api.c\_query\_op\_about, ctx\_api.c\_query\_op\_rt или ctx\_api.c\_query\_op\_syn.

**p\_thes\_name** - тезаурус, для которого выполняется функция. По умолчанию - 'default'. Игнорируется в режиме 'keyword'.

**p\_refine\_on** - флаг включения режима уточнения контекста запроса по весовым коэффициентам тематики на основе расширения токенов по иерархии ВТ вплоть до корневых суперкатегорий. В результирующей строке остаются только токены доминирующей тематики. Незначимые согласно структуре тезауруса токены исключаются из результирующей строки. Значение по умолчанию - ctx\_api.c\_refine\_off (отключено). Режим действует только в режиме 'concept', в режиме 'keyword' флаг игнорируется.

**p\_exp\_detail\_on** флаг включения режима расширения тематики запроса (режим сходен по своему действию с функцией NT). При включении режима выбираются такие уровни подфункций bt/nt результирующей строки, на которых иерархия тезауруса имеет не менее чем ctx\_api.c\_nt\_terms (равное 5) терминов NT для каждого токена исходной строки. Данный режим позволяет расширить поиск документов, выборочно расширив семантические подсети в узловых точках (токенах) до заданного количества терминов NT.

**Важно:**

- 1) данный режим работает только в режиме 'concept'.
- 2) Данный флаг является взаимоисключающим с флагом **p\_refine\_on**.
- 3) При задании данного флага параметр **p\_expansion\_level** игнорируется.
- 4) Данная функциональность работает только в режимах **p\_query\_opt** равных ctx\_api.c\_query\_op\_bt, ctx\_api.c\_query\_op\_nt (предназначен прежде всего для режима ctx\_api.c\_query\_op\_nt). Значение по умолчанию - ctx\_api.c\_exp\_detail\_off.

Функция может вызываться из SQL (при позиционном задании аргументов) и из PL/SQ (при задании аргументов по значению).

Примеры:

```
SQL> rem Тест 1. Вызов парсера по умолчанию, режим KEYWORD
SQL> select ctx_api.search_string_parser('сунны алмаз молитва животные бог
пассатижи','keyword') as "Parsed"
  2 from dual;
```

Parsed

```
-----
{сунны} and {алмаз} and {молитва} and {животные} and {бог} and {пассатижи}
```

Elapsed: 00:00:00.05

```
SQL> rem Тест 2. Вызов парсера по умолчанию, режим CONCEPT
SQL> select ctx_api.search_string_parser ('сунны алмаз молитва животные бог
пассатижи','concept') as "Parsed"
  2 from dual;
```

Parsed

```
-----
about({сунны}) and about({алмаз}) and about({молитва}) and about({животные}) and
about({бог}) and about({пассатижи})
```

Elapsed: 00:00:00.06

```
SQL> rem Тест 3. Тест уточнения тематики - вызов из SQL
SQL> select ctx_api.search_string_parser ('сунны алмаз молитва животные бог
пассатижи','concept','and','nt',1,'default',1) as "Parsed"
  2 from dual;
```

Parsed

```
-----
nt({сунны}) and nt({молитва}) and nt({бог})
```

Elapsed: 00:00:00.16

```
SQL> rem Тест 4. Тест уточнения тематики - вызов из PL/SQL
SQL> declare
  2   v_out varchar2(32767);
  3   begin
  4     v_out := ctx_api.search_string_parser('сунны алмаз молитва животные бог
пассатижи','concept', p_refine_on=>ctx_api.c_refine_on);
  5     dbms_output.put_line(v_out);
  6   end;
  7   /
about({сунны}) and about({молитва}) and about({бог})
```

PL/SQL procedure successfully completed.

Elapsed: 00:00:00.17

```
SQL> rem Тест 5. Вызов парсера по умолчанию, PL/SQL, режим CONCEPT, уточнение включено
SQL> declare
  2   v_out varchar2(32767);
  3   begin
  4     v_out := ctx_api.search_string_parser('сунны пассатижи пила пинцет
тиски','concept', p_refine_on=>ctx_api.c_refine_on);
  5     dbms_output.put_line(v_out);
  6   end;
  7   /
about({пассатижи}) and about({пила}) and about({пинцет}) and about({тиски})
```

PL/SQL procedure successfully completed.

Elapsed: 00:00:00.05

```
SQL> rem Тест 6. Проверка корректности действия уточнения тематики - все слова из
одной категории.
SQL> declare
  2   v_out varchar2(32767);
```

```

3  begin
4    v_out := ctx_api.search_string_parser('сунны молитва бог','concept',
p_refine_on=>ctx_api.c_refine_on);
5    dbms_output.put_line(v_out);
6  end;
7  /
about({сунны}) and about({молитва}) and about({бог})

```

PL/SQL procedure successfully completed.

Elapsed: 00:00:00.06

SQL> rem Тест 7. Проверка корректности действия уточнения категории слова в иерархическом расширительном запросе

SQL> rem NT/VT. При включении уточнения категории (p\_exp\_detail\_on) значение параметра p\_expansion\_level

SQL> rem ИГНОРИРУЕТСЯ!

SQL> declare

```

2    v_out varchar2(32767);
3  begin

```

```

4    v_out := ctx_api.search_string_parser('сунны молитва бог динозавр','concept',

```

```

5                                     p_query_opt=>ctx_api.c_query_op_nt,
6                                     p_refine_on=>ctx_api.c_refine_off,

```

```

7                                     p_exp_detail_on=>ctx_api.c_exp_detail_on);
8    dbms_output.put_line(v_out);
9  end;

```

```

10 /

```

```

nt({сунны},1) and nt({молитва},1) and nt({бог},1) and nt({динозавр},2)

```

PL/SQL procedure successfully completed.

При вызове функции через WSDL указываются все аргументы, причем константы необходимо задавать значениями:

Click [here](#) for a complete list of operations.

## searchStringParser

### Test

To test the operation using the HTTP GET protocol, click the 'Invoke' button.

Parameter	Type	Value
param0	string	<input type="text" value="сунны алмаз молитва животные бог пассатижи"/>
param1	string	<input type="text" value="concept"/>
param2	string	<input type="text" value="and"/>
param3	string	<input type="text" value="nt"/>
param4	decimal	<input type="text" value="5"/>
param5	string	<input type="text" value="default"/>
param6	decimal	<input type="text" value="1"/>
param7	decimal	<input type="text" value="0"/>

Результат вызова функции через WSDL:



```
<?xml version="1.0" encoding="UTF-8" ?>
= <SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema">
= <SOAP-ENV:Body>
  = <ns1:searchStringParserResponse xmlns:ns1="http://yvoinov.com/ctx_api.wsdl" SOAP-
    ENV:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/">
    <return xsi:type="xsd:string">nt({сунны},5) and nt({молитва},5) and
      nt({6or},5)</return>
    </ns1:searchStringParserResponse>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

**Замечание.** Текущая версия парсера имеет несколько известных проблем.

- p\_thes\_name не должно содержать символов «\_». Они будут удалены при выполнении, что приведет к исключению.
- Если определен параметр p\_exp\_detail\_on, равный 1 (ctx\_api.c\_exp\_detail\_on), значение p\_expansion\_level игнорируется.
- Если p\_query\_opt равен 'about', 'syn' или 'rt', или в случае если p\_query\_mode = 'keyword', параметр p\_exp\_detail\_on будет проигнорирован.
- Некоторые взаимоисключающие сочетания формальных параметров приводят к выводу пустой результирующей строки в результате перехвата возникающего исключения. Разработчику следует быть внимательным при задании параметров парсера.

При нормальном выполнении операций парсинга, равно как и при возникновении исключений, происходит обязательное освобождение памяти, используемой парсером. Это делает практически невозможными утечки памяти.

### Thesaurus content API

```
function term_counter (p_thes_name in varchar2 default 'default')
return number deterministic;
```

- Функция возвращает количество уникальных терминов для заданного тезауруса. Если заданный тезаурус не загружен, возвращается исключение ORA-20150.

Аргументы:

**p\_thes\_name** - тезаурус, для которого выполняется функция. По умолчанию - 'default'.

Функция может быть вызвана из SQL и PL/SQL.

Примеры:

```
select ctx_api.term_counter('default') from dual;
```

```
CTX_API.TERM_COUNTER('DEFAULT')
-----
                        71000
```

```
select ctx_api.term_counter('english') from dual;
```

```
CTX_API.TERM_COUNTER('ENGLISH')
-----
                        86886
```

При вызове функции через WSDL указывается единственный аргумент:

Click [here](#) for a complete list of operations.

### termCounter

### Test

To test the operation using the HTTP GET protocol, click the 'Invoke' button.

Parameter	Type	Value
param0	string	<input type="text" value="default"/>



Результат вызова функции через WSDL:

```
<?xml version="1.0" encoding="UTF-8" ?>
= <SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  = <SOAP-ENV:Body>
    = <ns1:termCounterResponse xmlns:ns1="http://yvoinov.com/ctx_api.wsdl" SOAP-
      ENV:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/">
      <return xsi:type="xsd:decimal">71000</return>
    </ns1:termCounterResponse>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

```
procedure thes_loaded (p_ths_list out ctx_api.thes_tab);
```

- Процедура возвращает список загруженных тезаурусов. Если не загружен ни один тезаурус, возвращается исключение ORA-20154.

Аргументы:

**p\_ths\_list** содержит массив возвращаемых значений (терминов) типа ctx\_api.thes\_tab.

Процедура может быть вызвана только из PL/SQL.

Пример:

```
declare
  xtab ctx_api.thes_tab;
  i number;
begin
  ctx_api.thes_loaded(xtab);
  if xtab.count > 0 then dbms_output.put_line('Has '||xtab.count||' thesauri');
end if;
  for i in 1..xtab.last loop
    dbms_output.put_line('Thes '||i||': '||xtab(i));
  end loop;
end;
/
```

```
Has 2 thesauri
Thes 1: ENGLISH
Thes 2: DEFAULT
```

PL/SQL procedure successfully completed.

При вызове процедуры через WSDL параметры не задаются:

Адрес: <http://blade/wSDL/ctxapi?operation=thesLoaded>

Click [here](#) for a complete list of operations.

## thesLoaded

### Test

To test the operation using the HTTP GET protocol, click the 'Invoke' button.

Parameter	Type	Value
		<input type="button" value="Invoke"/>

Результат вызова процедуры через WSDL:

```
<?xml version="1.0" encoding="UTF-8" ?>
= <SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  = <SOAP-ENV:Body>
    = <ns1:thesLoadedResponse xmlns:ns1="http://yvoinov.com/ctx_api.wsdl" SOAP-
      ENV:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/">
      = <return xmlns:ns2="http://yvoinov.com/ctx_api.xsd"
        xsi:type="ns2:yvoinov_com_ctx_apiUser_thesLoaded_Out">
        = <pthslistOut xsi:type="ns2:yvoinov_com_CtxApiThesTab">
          = <array xmlns:ns3="http://schemas.xmlsoap.org/soap/encoding/"
            xsi:type="ns3:Array" ns3:arrayType="xsd:string[1]">
            <item xsi:type="xsd:string">DEFAULT</item>
          </array>
        </pthslistOut>
      </return>
```

```
</ns1:thesLoadedResponse>  
</SOAP-ENV:Body>  
</SOAP-ENV:Envelope>
```

**Исключения**

Пакет CTX\_API использует 5 определенных разработчиком исключений Коды исключений приведены в таблице 2.

Таблица 2

Код ошибки	Причина	Устранение ошибки
<b>ORA-20150</b>	Ошибка Oracle Text. Заданный в подпрограмме тезаурус не существует.	Исправьте имя тезауруса на правильное и повторите вызов. При необходимости проверьте имена тезаурусов при помощи процедуры <code>ctx_api.thes_loaded</code>
<b>ORA-20151</b>	Заданная фраза не существует в тезаурусе или не загружен заданный тезаурус.	Исправьте имя тезауруса на правильное и повторите вызов. При необходимости проверьте имена тезаурусов при помощи процедуры <code>ctx_api.thes_loaded</code> . Если имя тезауруса указано верно, значит фраза действительно отсутствует.
<b>ORA-20152</b>	Заданная фраза xxx имеет гомографы.	Подпрограмма обнаружила гомографы термина (фразы) xxx при отсутствии квалификаторов. Получите все квалифицированные гомографы при помощи процедуры <code>ctx_api.get_qualifiers</code> , выберите нужный и задайте квалифицированную фразу, после чего повторите вызов.
<b>ORA-20154</b>	В систему не загружен ни один тезаурус.	Приобретите и загрузите иерархический тезаурус. Затем повторите попытку вызова.
<b>ORA-20155</b>	Произошла ошибка ORA-xxxxxx.	Смотрите код ошибки в Oracle Error Messages and Codes