CTX_API package. Description, procedures and functions

Version 1.0.0.5

CTX_API package. Description, procedures and functions
Version 1.0.0.5

Introduction

The CTX_API programming interface is designed to process search queries to a search engine based on Oracle Text. This interface uses a hierarchical thesaurus of the Russian language[1]. It allows you to both fully process the search query and perform various lower-level functions, called, for example, from the application interface (implementing advanced search functions, etc.).

The interface is written in PL/SQL and is also implemented as a WSDL service that uses the standard SOAP protocol to organize B2B interaction[2].

CTX_API implements the function of the high-speed universal parser and auxiliary calls of the CTXSYS.CTX_THES package. The main function of the interface is to perform intelligent parsing of search strings (queries) based on the metadata of the loaded thesauri for subsequent query execution by the Oracle Text search engine[3].

The specificity of the CTX_API interface is such that it is not tied to the use of one particular natural language. The package is capable of working with any language that uses words separated by spaces. This makes it possible to use any number of thesauri at the same time[4].

The package is not tied to a particular target hardware and software platform and can be used on all platforms supported by Oracle.

Supported versions of Oracle DBMS: **> 9.2.0.3**[5]

[1] Can use any number of hierarchical thesauri in any language. The search query parser is able to function even in the absence of loaded thesauri (in KEYWORD mode).

[2] B2B - Business-To-Business. A common abbreviation for Service-to-Service interaction. It implies the provision of services over the global network of one business to another.

[3] And others made in accordance with ISO-2788,ANSI Z39.19 standards.

[4] Theoretically, all, except for those using hieroglyphic writing.

[5] All versions of Oracle starting from 9.2.0.3 containing the new Oracle Text implementation are supported.

CTX_API Installation Guide

API installation can be done from the client (using Oracle Client/SQL*Net > 8.1.7) or from the server.

The prerequisites for installation are the presence of an installed Oracle Server (Standard or Enterprise Edition[6]) with the Oracle Text option installed and configured (manually or via netca/netmgr) SQL*Net.

******* The installation is performed with SYS user rights.

For installation, it is necessary (depending on the target platform) to run the wrapper script **inst_api.bat/inst_api.sh** .

When executing the installation script, you are asked :

- The name of the target schema where the API will be installed;
- SYS user password;
- ORACLE_SID (SERVICE_NAME) of the target database[7]

The target schema must be created with the following rights:

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

After executing the setup scripts, the target schema will be given - the right to execute the CTXSYS.CTX_THES package, the CTXAPP role, and the right to select from the CTXSYS.CTX_THESAURI and CTXSYS.CTX_THES_PHRASES views.

API routines are executed with CURRENT_USER rights.

The target schema must remain unlocked and a quality password must be set for its security.

Checking the correct installation of the API is to run the following commands :

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

The request must return the API version number.

Note: The version and search_string_parser[8] functions are functional even without thesauri loaded. The rest of the API functions require at least one hierarchical thesaurus loaded into the database to be successful[9]. Otherwise, **ORA-20154:** No thesaurus found and **ORA-20150:** Oracle Text error are thrown. Possible specified thesaurus not loaded.

---

[6] In theory, support for Oracle Express Edition (XE) is possible, but using CTX_API on this version is not practical, except for application development. However, due to the serious functional limitations of Oracle XE, this installation is not recommended.

[7] Before installation, it is desirable to make sure using the TNSPING utility that the service name/database instance name is accurate and that it is available.

[8] Only in KEYWORD mode.

[9] **The hierarchical thesaurus of the Russian language** is a commercial product.

CTX_API package. Description, procedures and functions
Version 1.0.0.5

List of procedures and functions of the CTX_API package and their brief description

The subroutines of the CTX_API package are logically divided into three functional groups : API version, Thesaurus CTX API, Thesaurus content API.

Functional descriptions of the package subroutines are given in the table 1.

Таблица 1

Package Subroutine	Functional description
function **version**	Returns the version number of the interface. Used to check interoperability.
function **phrase_exists**	Returns true if the given term exists in the given thesaurus. Used to check if a term exists in the thesaurus.
function **phrase_relation_exists**	The function returns the NT/BT level at which, relative to the given term, there are more than c_nt_terms of sibling NT terms in the thesaurus hierarchy tree (c_nt_terms defaults to 5).
function **search_expansion_term**	The function returns the parent term BT of the subcategory for which, relative to the given term, there are more than c_nt_terms of sibling NT terms in the thesaurus hierarchy tree (c_nt_terms defaults to 5).
function **has_homographs**	The function checks the existence of homographs for the given term. Returns true if there is at least one homograph, false otherwise.
procedure **get_qualifiers**	The procedure returns the qualifier for the given term, if the qualifier exists. Designed to define the parent subcategory for homographs in extension queries
function **get_note**	The function returns a Scope Note (SN) - a comment for the given term in the given thesaurus. If SN does not exist, an empty string is returned. If the term does not exist or the thesaurus is not loaded, an ORA-20151 exception is returned.
function **get_bt**	The function returns the subcategory for the given term (single term). If the term is not in the thesaurus, the function returns only that term. If a term has homographs, but none were specified when called, an ORA-20152 exception is returned. If the term is given with a qualifier, the BT subcategory of the given level is returned.
procedure **get_bt**	The procedure returns **ALL** BT subcategories for the given term (BT subtrees). If the term is not in the thesaurus, the procedure returns only that term. If a term has homographs but no qualifier is given when the procedure is called, all BT subcategories (BT subtrees) for each homograph are returned, with each subtree starting with the given term with the given branch qualifier. Subtrees are displayed in reverse order (i.e., the given term is at the top, subsequent BT subcategories go in reverse order to the level specified when calling the procedure, or to the top of the hierarchy tree if the term is at a level less than the given value of the hierarchy level). If a term has homographs and a qualifier

	is given when the procedure is called, the procedure returns only the BT hierarchy subtree of the given term.
procedure **get_nt**	The procedure returns the NT terms for the given term. If the term is not in the thesaurus, an ORA-20151 exception is returned. If the term has homographs but no qualifier is given, an ORA-20152 exception is returned. If a term has homographs and a qualifier is given, a subtree of NT terms of the given extension level is returned. If the term is at the lowest level of the hierarchy (has no NT terms), only the given term is returned.
procedure **get_ntp**	The procedure returns the NTP terms for the given term. If the term is not in the thesaurus, an ORA-20151 exception is returned. If the term has homographs but no qualifier is given, an ORA-20152 exception is returned. If a term has homographs and a qualifier is given, a subtree of NTP terms of the given extension level is returned. If the term is at the lowest level of the hierarchy (has no NTP terms), only the specified term is returned.
procedure **get_rt**	The procedure returns the associated RT terms for the given term. If the term is not in the thesaurus, an ORA-20151 exception is returned. If the term has homographs but no qualifier is given, an ORA-20152 exception is returned. If a term has homographs and a qualifier is given, all of its RT associations are returned. If the term has no RT terms, only the specified term is returned.
procedure **get_syn**	The procedure returns the SYNs for the given term. If the term is not in the thesaurus, an ORA-20151 exception is returned. If the term has homographs but no qualifier is given, an ORA-20152 exception is returned. If a term has homographs and a qualifier is given, all of its synonyms are returned. If the term has no synonyms, only the specified term is returned.
function **search_string_parser**	Function of a universal high-speed parser. Supports multiple thesauri, works without thesauri in KEYWORD mode (default), supports basic relationships of ISO-2788 and ANSI Z39.19 standards. Also supports all logical operands, phrases and homograph qualifiers.
function **term_counter**	The function returns the number of unique terms for the given thesaurus. If the given thesaurus is not loaded, an ORA-20150 exception is returned.
procedure **thes_loaded**	The procedure returns a list of loaded thesauri. If no thesaurus is loaded, an ORA-20154 exception is returned.

Using CTX_API

API version

```
function version return varchar2 deterministic;
```

- Returns the version number of the interface. Used to check interoperability.

The function can be called from both PL/SQL and SQL.

Example:

```
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
```

The WSDL interface returns the value of the given function in the format of a SOAP message:

```
<?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

Constants, Variables, and Package Types:

```
-- 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;
```

The c_nt_terms constant specifies the number of NT terms for the subcategory at which the **search_expansion_level** and **search_expansion_term** expansion functions should terminate. This constant is also used by the **search_string_parser** parser in p_exp_detail_on in number default ctx_api.c_exp_detail_on mode.

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

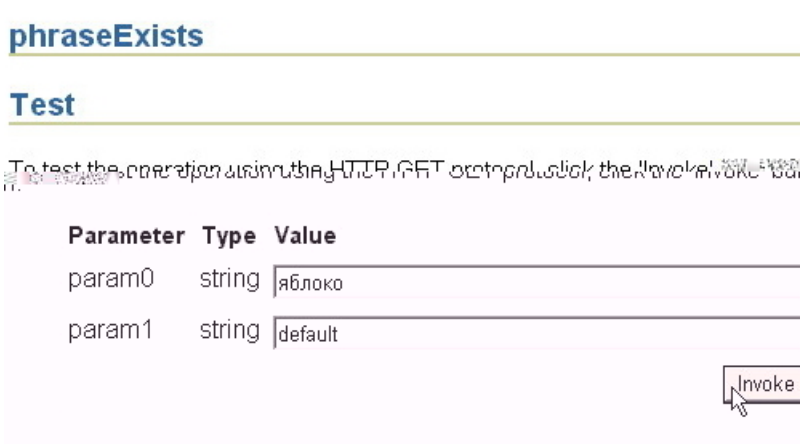
- The function checks if the given term exists in the thesaurus.
  Arguments:
  p_phrase used to specify a term, may contain a qualifier.
  P_thes_name - thesaurus for which the function is executed. The default is 'default'.

The function can only be called from within PL/SQL.

Example:

```
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 calls require all arguments to be specified explicitly:



The result of calling the function via 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;
```

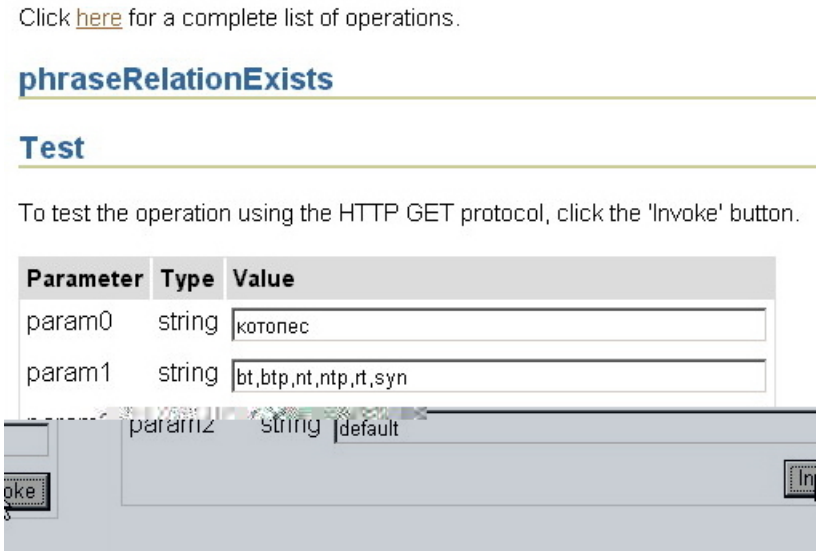
- The function checks if the given relation(s) exists in the thesaurus for the given term.
  Arguments:
  p_phrase used to specify a term, may contain a qualifier.
  p_relation is a string list of relationships against which the term is tested. By default 'bt,btp,nt,ntp,rt,syn'[10].
  P_thes_name – thesaurus for which the function is executed. The default is 'default'.

The function can only be called from within PL/SQL.

Example:

```
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 calls require all arguments to be specified explicitly:



The result of calling the function via WSDL:

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

[10] The default value is a list of the main relations of **the hierarchical thesaurus of the Russian language.**

CTX_API package. Description, procedures and functions
Version 1.0.0.5


```
- <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: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;
```

- The function returns the NT/BT level at which, relative to the given term, there are more than c_nt_terms of sibling NT terms in the thesaurus hierarchy tree (c_nt_terms defaults to 5).
  Arguments:
  p_phrase used to specify a term, may contain a qualifier.
  P_thes_name - thesaurus for which the function is executed. The default is 'default'.

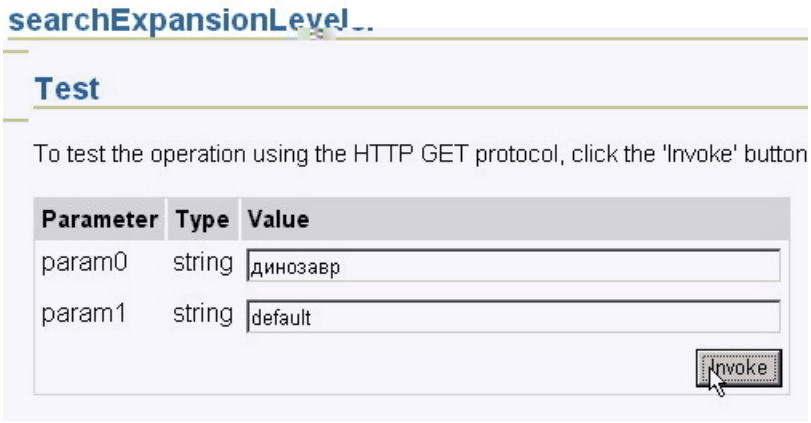
The function can be called from PL/SQL and SQL.

Example:

```
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.
```

When called via WSDL, all arguments are specified:

Click here for a complete list of operations.



The result of calling the function via 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;
```

- The function returns the parent term BT of the subcategory for which, relative to the given term, there are more than c_nt_terms of sibling NT terms in the thesaurus hierarchy tree (c_nt_terms defaults to 5).
  Arguments:
  p_phrase used to specify a term, may contain a qualifier.
  p_thes_name – thesaurus for which the function is executed. The default is 'default'.

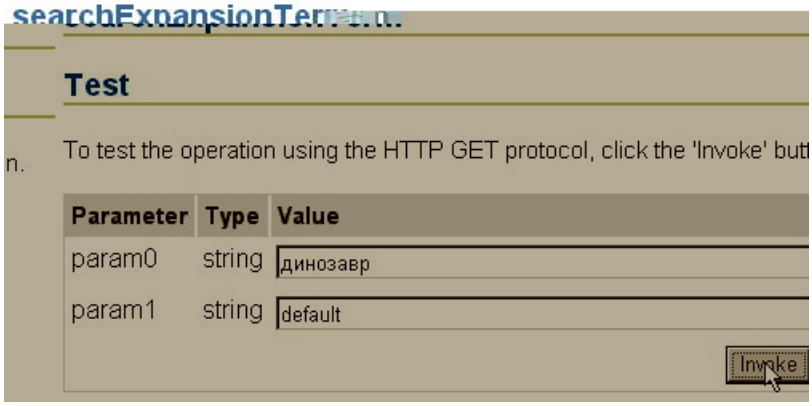
The function can be called from PL/SQL and SQL.

Example:

```
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.
```

When called via WSDL, all arguments are specified:

Click here for a complete list of operations.



The result of calling the function via 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;
```

- The function checks the existence of homographs for the given term. Returns true if there is at least one homograph, false otherwise.
  Arguments:
  p_phrase used to specify a term, may contain a qualifier.
  p_thes_name – thesaurus for which the function is executed. The default is 'default'.

The function can only be called from within PL/SQL.

Example:

```
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.
```

When called via WSDL, all arguments are specified[11]:

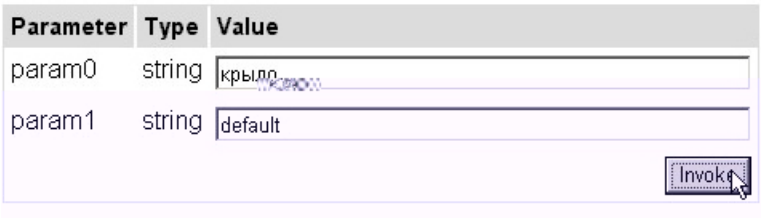
[11] When called as part of a B2B service, arguments are passed via URL:
http://<hostname>/wsdl/ctxapi?invoke=hasHomographs&param0=крыло&param1=default

Click here for a complete list of operations.

hasHomographs

Test

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



The result of calling the function via 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');
```

- The procedure returns the qualifier for the given term, if the qualifier exists. Designed to define the parent subcategory for homographs in extension queries.
  Arguments:
  p_qualifiers contains an array of return values (terms) of type ctx_api.term_tab.
  p_phrase used to specify a term, may contain a qualifier.
  p_thes_name – thesaurus for which the function is executed. The default is 'default'.

The procedure can only be called from within PL/SQL.

Example:

```
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
```

When called via WSDL, all arguments are specified:

CTX_API package. Description, procedures and functions
Version 1.0.0.5

Click here for a complete list of operations.

getQualifiers

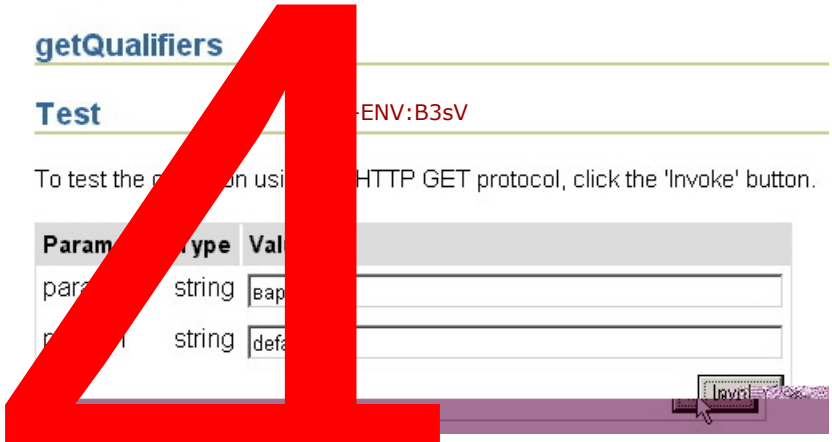
Test

_ <SOAP-ENV:Envel xmlns:SOAP-="

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

Parameter	Type	Value
param0	string	якорь
param1	string	default

http://schemas.xmlsoap.org/soap/envelфOsdl



Click here for a complete list of operations.

getQualifiers

Test

-ENV:B3sV

To test the o n usi HTTP GET protocol, click the 'Invoke' button.

Param ype	Val
para string	вар
n string	defa

Invol

Results of proce calls via WSDL:

```
<?xml version="1.0" enco "UTF-8"?>
- <SOAP-ENV:Envelope xm AP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xmlns:xsd="http://www.w3.org/2001/XMLSchema" inst2vhng/
  - <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>
  </
```

b

b

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>

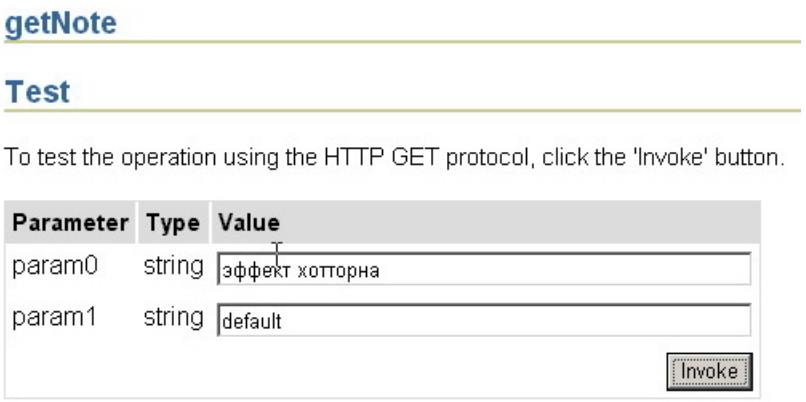
```
function get_note (p_phrase in varchar2,
                   p_thes_name in varchar2
                   default 'default') return varchar2
                   deterministic;
```

- The function returns a Scope Note (SN) for the given term if SN is defined (exists). It is intended to extract the comment (SN) if it is set for the term. If the term does not exist or the thesaurus is not loaded, an ORA-20151 exception is returned.

  Arguments:
  p_phrase used to specify a term, may contain a qualifier.
  p_thes_name – thesaurus for which the function is executed. The default is 'default'.

The procedure can only be called from within PL/SQL.

When called via WSDL, all arguments are specified:



Results of procedure calls via 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;
```

- The function returns the subcategory for the given term (single term). If the term is not in the thesaurus, the function returns only that term. If a term has homographs, but none were specified when called, an ORA-20152 exception is returned. If the term is given with a qualifier, the BT subcategory of the given level is returned.

  Arguments:
  p_phrase used to specify a term, may contain a qualifier.
  p_level hierarchy level in BT relationships. The default value is 1.
  p_thes_name – thesaurus for which the function is executed. The default is 'default'.

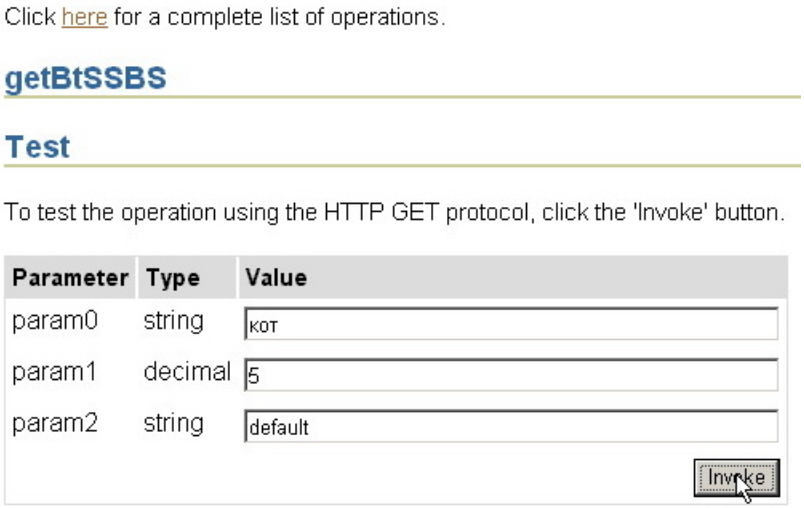
The function can be called from PL/SQL and SQL.

Examples:

```
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)
```

When called via WSDL, all arguments are specified.

***** ATTENTION!** Note that in the WSDL interface, this function named **getBtSSBS**!



The result of calling the function via 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>
```

As follows from this example, the function returns 5 levels up the thesaurus BT hierarchy down to the **BIOLOGY** subcategory, and returns directly the level 5 subcategory for the given term.

```
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');
```

- The procedure returns **ALL** BT subcategories for the given term (BT subtrees). If the term is not in the thesaurus, the procedure returns only that term. If a term has homographs but no qualifier is given when the procedure is called, all BT subcategories (BT subtrees) for each homograph are returned, with each subtree starting with the given term with the given branch qualifier. Subtrees are displayed in reverse order (i.e., the given term is at the top, subsequent BT subcategories go in reverse order to the level specified when calling the procedure, or to the top of the hierarchy tree if the term is at a level less than the given value of the hierarchy level). If a term has homographs and a qualifier is given when the procedure is called, the procedure returns only the BT hierarchy subtree of the given term.

  Arguments:
  p_bt contains an array of return values (terms) of type ctx_api.term_tab.
  p_phrase used to specify a term, may contain a qualifier.
  p_level hierarchy level in BT relationships. The default value is 1.
  p_thes_name – thesaurus for which the function is executed. The default is 'default'.

The procedure can only be called from within PL/SQL.

Examples:

```
SQL> declare
  2    xtab ctx_api.term_tab;
  3  begin
  4    -- Термин с гомографами - выводятся оба субдерева BT 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.
```

When called via WSDL, all arguments are specified:

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

Invoke

The result of calling the procedure via 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>
```

As you can see from this example, the procedure returns both qualified BT relationship hierarchical subtrees for the term DIAMOND up to a maximum of 5 levels up the thesaurus hierarchy.

```
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');
```

- The procedure returns the NT terms for the given term. If the term is not in the thesaurus, an ORA-20151 exception is returned. If the term has homographs but no qualifier is given, an ORA-20152 exception is returned. If a term has homographs and a qualifier is given, a subtree of NT terms of the given extension level is returned. If the term is at the lowest level of the hierarchy (has no NT terms), only the given term is returned.

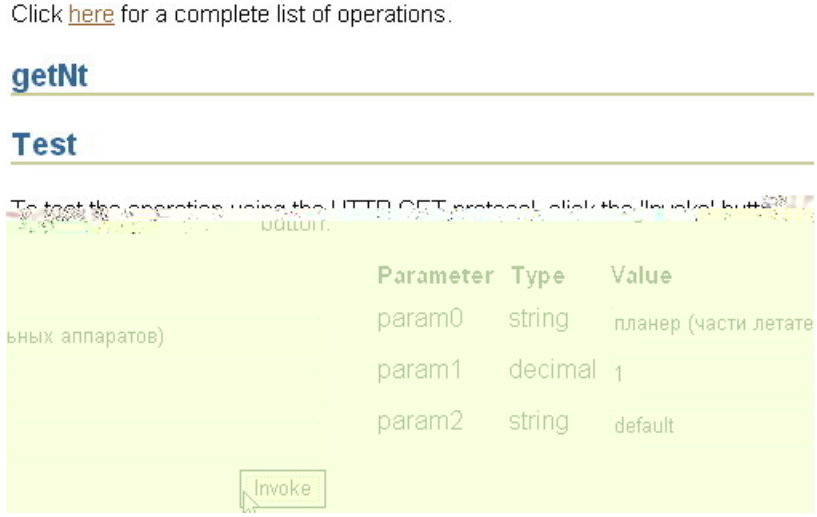
  Arguments:
  p_nt contains an array of return values (terms) of type ctx_api.term_tab.
  p_phrase used to specify a term, may contain a qualifier.
  p_level hierarchy level in BT relationships. The default value is 1.
  p_thes_name – thesaurus for which the function is executed. The default is 'default'.

The procedure can only be called from within PL/SQL.

Example:

```
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
```

When called via WSDL, all arguments are specified:



The result of calling the procedure via 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');
```

- The procedure returns the NTP terms for the given term. If the term is not in the thesaurus, an ORA-20151 exception is returned. If the term has homographs but no qualifier is given, an ORA-20152 exception is returned. If a term has homographs and a qualifier is given, a subtree of NTP terms of the given extension level is returned. If the term is at the lowest level of the hierarchy (has no NTP terms), only the specified term is returned.

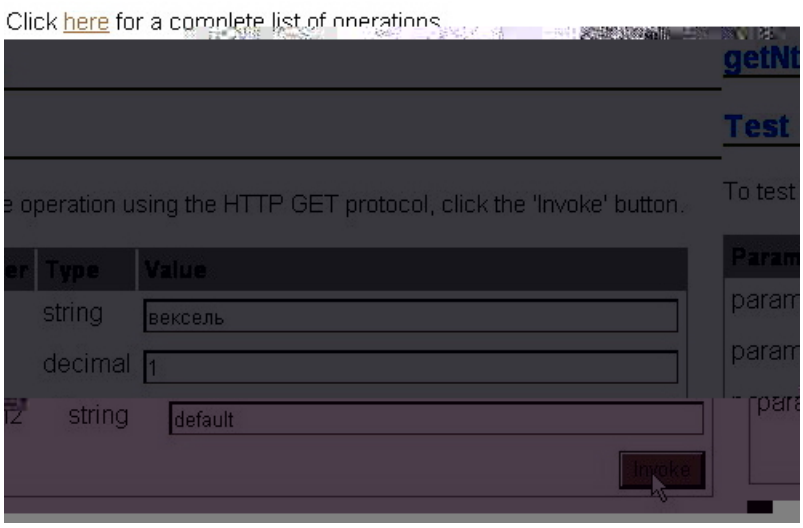
  Arguments:
  p_ntp contains an array of return values (terms) of type tx_api.term_tab.
  p_phrase used to specify a term, may contain a qualifier.
  p_level hierarchy level in BT relationships. The default value is 1.
  p_thes_name – thesaurus for which the function is executed. The default is 'default'.

The procedure can only be called from within PL/SQL.

Example:

```
  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.
```

When called via WSDL, all arguments are specified:



The result of calling the procedure via 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');
```

- The procedure returns the associated RT terms for the given term. If the term is not in the thesaurus, an ORA-20151 exception is returned. If the term has homographs but no qualifier is given, an ORA-20152 exception is returned. If a term has homographs and a qualifier is given, all of its RT associations are returned. If the term has no RT terms, only the specified term is returned.

  Arguments:
  p_rt contains an array of return values (terms) of type tx_api.term_tab.
  p_phrase used to specify a term, may contain a qualifier.
  p_thes_name – thesaurus for which the function is executed. The default is 'default'.

The procedure can only be called from within PL/SQL.

Example:

```
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
```

When called via WSDL, all arguments are specified:

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	мифология
param1	string	default

Invoke

The result of calling the procedure via 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');
```

-

```
            <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;
```

- Function of a universal high-speed parser. Supports multiple thesauri, works without thesauri in KEYWORD mode (default), supports basic relationships of ISO-2788 and ANSI Z39.19 standards. Also supports all logical operands, phrases and homograph qualifiers.

  Arguments:
  p_search_string – пThe search string passed to the parser for processing. Must be limited to 4000 characters (maximum capacity of internal buffers).
  p_query_mode defines the parser's mode of operation ('keyword' – search by keywords or 'concept' – search by topic based on thesaurus). The default value is 'keyword' (in this mode, the parser can work without a thesaurus). The parameter determines the character of parser output string conversion.
  p_logical_op – boolean operand applied to the tokens of the output string. Has two meanings - 'and' or 'or'. The default is 'and'.
  p_query_opt – output line generation mode in 'concept' mode. One of five values is set: 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. The default value is ctx_api.c_query_op_about. The constants define the Oracle Text functions that will be used to form the final string passed to the search engine for execution. Ignored in 'keyword' mode.
  p_expansion_level – hierarchy level in bt/nt extension functions in 'concept' mode. The default value is 1. Ignored in 'keyword' mode and in 'concept' mode if p_query_opt is equal to ctx_api.c_query_op_about, ctx_api.c_query_op_rt or ctx_api.c_query_op_syn.
  p_thes_name – thesaurus for which the function is executed. The default is 'default'. Ignored in 'keyword' mode.
  p_refine_on – flag for enabling the query context refinement mode by subject weights based on the expansion of tokens along the BT hierarchy up to the root supercategories. Only the tokens of the dominant theme remain in the resulting string. Tokens that are insignificant according to the structure of the thesaurus are excluded from the resulting string. The default value is ctx_api.c_refine_off (disabled). The mode is valid only in 'concept' mode, in 'keyword' mode the flag is ignored.
  p_exp_detail_on - flag for enabling the query topic extension mode (the mode is similar in its effect to the NT function). When the mode is enabled, the levels of bt/nt subfunctions of the resulting string are selected at which the thesaurus hierarchy has at least ctx_api.c_nt_terms (equal to 5) NT terms for each source string token. This mode allows you to expand the search for documents by selectively expanding semantic subnets at key points (tokens) up to a specified number of NT terms.
  Important note:
  1) This flag only works in 'concept' mode.
  2) This flag is mutually exclusive with the flag **p_refine_on**.
  3) When this flag is set, the parameter **p_expansion_level** will be ignored.
  4) This functionality works only in **p_query_opt** modes equal to ctx_api.c_query_op_bt, ctx_api.c_query_op_nt (intended primarily for ctx_api.c_query_op_nt mode). The default value is ctx_api.c_exp_detail_off.

The function can be called from SQL (when arguments are positionally specified) and from PL/SQ (when arguments are specified by value).

Examples:

```
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/BT. 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.
```

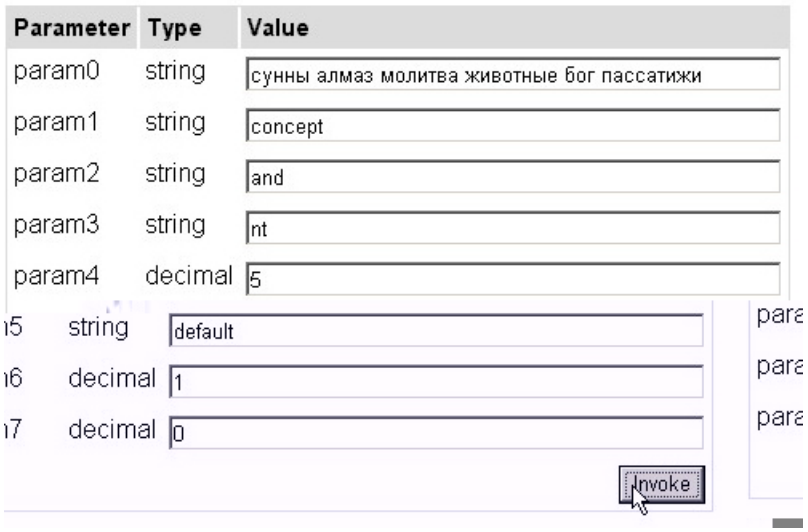
When calling a function through WSDL, all arguments are specified, and constants must be set with values:

Click here for a complete list of operations.

searchStringParser

Test

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



The result of calling the function via 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({бог},5)</return>
    </ns1:searchStringParserResponse>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

Note. The current version of the parser has several known issues.

- p_thes_name must not contain "_" characters. They will be removed on execution, resulting in an exception.
- If p_exp_detail_on is set to 1 (ctx_api.c_exp_detail_on), p_expansion_level is ignored.
- If p_query_opt is 'about','syn' or 'rt', or if p_query_mode = 'keyword', then p_exp_detail_on will be ignored.
- Some mutually exclusive combinations of formal parameters result in the output of an empty result string as a result of catching the resulting exception. The developer should be careful when setting parser parameters.

During the normal execution of parsing operations, as well as when exceptions occur, the memory used by the parser is necessarily freed. This makes memory leaks nearly impossible.

Thesaurus content API

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

- The function returns the number of unique terms for the given thesaurus. If the given thesaurus is not loaded, an ORA-20150 exception is returned.

  Arguments:
  p_thes_name – thesaurus for which the function is executed. The default is 'default'.

The function can be called from SQL and PL/SQL.

Examples:

```
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
```

When calling a function via WSDL, a single argument is specified:

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	default

Invoke

The result of calling the function via 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);
```

- The procedure returns a list of loaded thesauri. If no thesaurus is loaded, an ORA-20154 exception is returned.

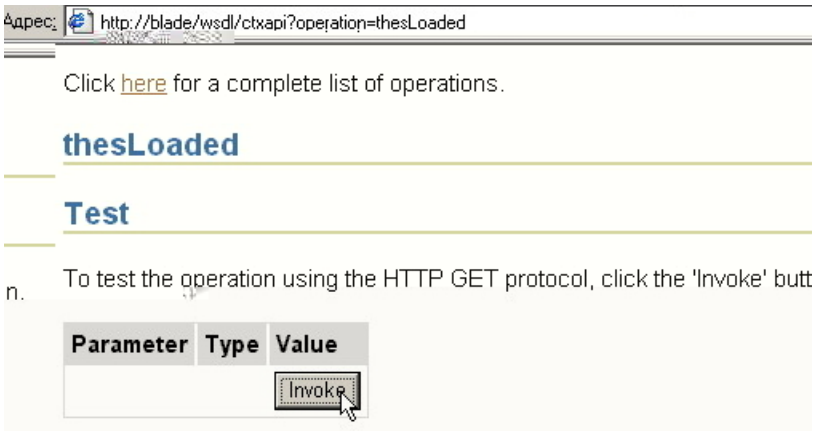
  Arguments:
  p_ths_list contains an array of return values (terms) of type ctx_api.thes_tab.

The procedure can only be called from within PL/SQL.

Example:

```
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.
```

When calling a procedure via WSDL, no parameters are set:



The result of calling the procedure via 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

The CTX_API package uses 5 developer-defined exceptions. The exception codes are shown in Table 2.

Таблица 2

Error code	Reason	Troubleshooting
ORA-20150	Oracle text error. The thesaurus specified in the routine does not exist.	Correct the thesaurus name to the correct one and retry the call. Check the thesauri names, if necessary, using the ctx_api.thes_loaded procedure
ORA-20151	The specified phrase does not exist in the thesaurus, or the specified thesaurus is not loaded.	Correct the thesaurus name to the correct one and retry the call. If necessary, check the thesauri names using the ctx_api.thes_loaded procedure. If the thesaurus name is correct, then the phrase is indeed missing.
ORA-20152	The given phrase xxx has homographs.	The subroutine found homographs of the term (phrase) xxx in the absence of qualifiers. Get all qualified homographs using the ctx_api.get_qualifiers procedure, select the one you want and set the qualified phrase, then repeat the call.
ORA-20154	No thesaurus is loaded into the system.	Purchase and load the hierarchical thesaurus. Then try the call again.
ORA-20155	An ORA-xxxxxx error has occurred.	Look at the Oracle Error Messages and Codes.