

## **\*\* OnyxCeph<sup>3</sup>™ Instruction \*\***

### ***How to request analysis results from OnyxCeph<sup>3</sup>?***

The Onyx module OnyxQuery can be used to query patients, images and results from OnyxCeph<sup>3</sup>. The results are recalculated based on current data and output as XML or CSV.

## **1. Start OnyxQuery**

The communication with OnyxQuery is done via a command line interface (CLI).

For query OnyxQuery is called with the appropriate parameters.

If OnyxQuery is already active on this computer, this will be detected, the request will be passed to the already running instance and answered by it.

If the log window is minimized, OnyxQuery continues to run as an icon in the taskbar.

When first started, OnyxQuery connects to the server configured in OnyxCeph. This is independent of whether a request is already passed or not.

If the user administration is active in OnyxCeph<sup>3</sup>, login and password are now requested.

## **2. Request information**

**OnyxQuery** supports several query modes. All of them have in common that the result is written to an XML file and/or a CSV file. Path and name of at least one result file must be passed with each call via the parameter `-XML=` or `-CSV=`.

The calling program has to make sure that the file name is valid and that the necessary write permissions are available.

Also, the calling program is responsible for deleting this file.

### **2.1. List of patients**

Patients can be queried with **OnyxQuery**.

#### **Syntax**

```
OnyxQuery.exe -XML="File name" -QueryMode=P -Client=ClientID
```

The transfer of ClientID is optional. If ClientID is not transferred, the patient is searched for in the HomeClient.

#### **Example**

```
OnyxQuery.exe -XML="D:\Res.xml" -QueryMode=P
```

#### **Result (abbreviated)**

```
<?xml version="1.0" encoding="UTF-8"?>
  <OnyxCeph-Patients>
    <Patient ID="10000" FirstName="Daniela" LastName="Demo" MiddleName="M."/>
    <Patient ID="10001" FirstName="Frank" LastName="Last"/>
    ****
  </OnyxCeph-Patients>
```

If the parameter `-PatID=PatientenID` is also specified, only the data of this patient will be exported.

## 2.2. Image findings of a patient

**OnyxQuery** can be used to query the available image datasets of a patient.

### Syntax

```
OnyxQuery.exe -XML="File name" -QueryMode=I -Client=ClientID -PatID=PatientenID
```

If ImageTypeID or ImageTypeCode is passed, only images of this type will be returned.

The transfer of ClientID is optional. If ClientID is not passed, the patient is searched for in the HomeClient.

### Example

```
OnyxQuery.exe -XML="D:\Res.xml" -QueryMode=I -PatID="10000"
```

### Result (abbreviated)

```
<?xml version="1.0" encoding="UTF-8"?>
  <OnyxCeph-Images PatID="10000">
    <Image ImgCode="10000INIT" Date="2006-04-05" ImageTypeID="36"
      ImageTypeName="Modell Wechselgebiss 3D" Kind="3D"
      ImageTypeCode="CAM3DD"/>
    <Image ImgCode="10000RX1S" Date="2006-04-05" ImageTypeID="36"
      ImageTypeName="Modell Bleibendes Gebiss 3D" Kind="3D"
      ImageTypeCode="CAM3DD"/>
    <Image ImgCode="100002VSS" Date="2007-07-04" ImageTypeID="34"
      ImageTypeName="Modell Bleibendes Gebiss 3D" Kind="3D"/>
      ImageTypeCode="CAP3D"/>
    ****
  </OnyxCeph-Images>
```

## 2.3. Query tracing types for an image type

In OnyxCeph<sup>3</sup>, image types can have several analysis types (tracing types). For example, a lateral photo analysis from the left, from the right and a frontal photo analysis can be performed on the FaceScan 3D image type.

The following call can be used to query the evaluation types (TracingTypeID) available for an image type (ImageTypeID or ImageTypeCode).

Alternatively, the ImageTypeCode parameter can also be used to select the image type.

### Syntax

```
OnyxQuery.exe -XML="File name" -QueryMode=T -ImageTypeID=ImageType
OnyxQuery.exe -XML="File name" -QueryMode=T -ImageTypeCode=ImageTypeCode
```

The transfer of ClientID is optional. If ClientID is not transferred, the patient is searched for in the HomeClient.

### Example

```
OnyxQuery.exe -XML="D:\Res.xml" -QueryMode=T -ImageTypeID=34
```

### Result

```
<?xml version="1.0" encoding="UTF-8"?>
  <OnyxCeph-TracingTypes ImageTypeID="34">
    <TracingType TracingTypeID="28" TracingTypeName="Modell 3D Milchgebiss"
      Kategorie="3D"/>
    <TracingType TracingTypeID="29" TracingTypeName="Modell 3D Wechselgebiss"
      Kategorie="3D"/>
    <TracingType TracingTypeID="30" TracingTypeName="Modell 3D bleibend"
      Kategorie="3D"/>
  </OnyxCeph-TracingTypes>
```

## 2.4. Request available analyses for the tracing type

In OnyxCeph<sup>3</sup> several analyses are available for each tracing type. The following call can be used to query the analyses available for a tracing type (TracingTypeID).

### Syntax

```
OnyxQuery.exe -XML="File name" -QueryMode=A -TracingTypeID=TracingType
```

### Example

```
OnyxQuery.exe -XML="D:\Res.xml" -QueryMode=A -TracingTypeID=30
```

### Result (abbreviated)

```
<?xml version="1.0" encoding="UTF-8"?>
  <OnyxCeph-Analyses TracingTypeID="30">
    <Analyse AnaCode="352696261" AnaName="Arcogrammetrie"/>
    <Analyse AnaCode="352665092" AnaName="Bogenlängendifferenz"/>
    <Analyse AnaCode="D7P7HAJIOP" AnaName="Bolton"/>
    <Analyse AnaCode="352351969" AnaName="Bolton - anterior"/>
    <Analyse AnaCode="353029178" AnaName="Bolton - overall"/>
    <Analyse AnaCode="XIA88T7QPG" AnaName="Bonn"/>
    ****
  </OnyxCeph-Analyses>
```

## 2.5. Request analysis results

The analyses in OnyxCeph<sup>3</sup> are identified via the AnaCode. The image dataset is selected via the ImgCode. This code corresponds to the one that can also be transferred from OnyxCeph<sup>3</sup> to other communication partners via VDDSmmedia. All values of the analysis are output, not calculated or not defined values are output empty.

### Syntax

```
OnyxQuery.exe -XML="File name" -QueryMode=R -ImgCode=ImgCode -AnaCode=AnaCode
```

### Example

```
OnyxQuery.exe -XML="D:\Res.xml" -QueryMode=R -ImgCode="10002VSS"
  -AnaCode="352665092"
```

### Result (abbreviated)

```
<?xml version="1.0" encoding="UTF-8"?>
  <OnyxCeph-Results Name="Demo, Daniela" PatID="10000" BirthDate="1994-03-13"
    ClientID="2350" ImgDate="2007-07-04" TracingTypeName="Modell 3D bleibend"
    TracingTypeID="30" AnaName="Bogenlängendifferenz" AnaCode="352665092"
    ImgCode="10002VSS" OnyxLink="-SelectImage=10002VSS -Client=1234">
    <resultitem name="ZBLo">
      <value unit="mm">78.1</value>
      <reference/>
      <referenceMin/>
      <referenceMax/>
      <deviation/>
      <interpretation/>
    </resultitem>
    <resultitem name="ZBSro">
      <value unit="mm">37.4</value>
      <reference unit="mm">39.0</reference>
      <referenceMin/>
      <referenceMax/>
      <deviation unit="mm">-1.7</deviation>
      <interpretation/>
    </resultitem>
```

```

<resultitem name="BLDro">
  <value unit="mm">1.7</value>
  <reference/>
  <referenceMin/>
  <referenceMax/>
  <deviation/>
  <interpretation/>
</resultitem>
****
</OnyxCeph-Results>

```

## 2.6 Retrieving the image data or the thumbnail of an image dataset

In QueryMode F for File, a copy of the image findings can be queried.

The parameter `-Option=` must be used to specify the format in which the data is to be saved.

- **2D/3D THUMB:** (default) The thumbnail (250 pixels, JPG) is saved.
- **2D** JPG, TIF, PNG, J2K or BMP: The cropped and rotated original image is saved in the specified format.
- **3D** STL, OBJ, PLY, ZPR, OFF oder DXF: The unchanged original objects (no planings, brackets or appliances) are saved in the requested format.
  - Objects with a textured surface are only taken over in OBJ
  - If the format supports only one object, all exportable objects are output together..

### Syntax

```

OnyxQuery.exe -ImgFile="File name" -QueryMode=F -ImgCode=ImgCode
               -Client=ClientID -Option="EXT"

```

The transfer of the ClientID is optional. If `-Client=` is not passed, the HomeClient is used.

The passing of `-Option` is optional. If option is not passed, "Tumb" is used.

When saving a thumbnail, the file extension is forced. If the extension given by `-ImgFile` is not correct, the correct extension is appended to the file name.

### Beispiel

The following call saves unchanged objects of a finding as `C:\Tmp\Img.obj`

```

OnyxQuery.exe -ImgFile=C:\Tmp\Img.obj -QueryMode=F -ImgCode="100002VSS"
               -Option="OBJ"

```

## 3. Output of the results as Comma-Separated Values (CSV)

The calculation results can be output in CSV format in addition to XLM format.

To do this, use the following parameter:

```
-CSV="File name"
```

The parameter `-CSV=` can be used instead of or together with `-XML=`. However, at least one result file must be specified. (except in `-QueryMode=F`)

The desired format of the CSV file can be modified using the "CSV" button at the bottom left of the log window. It is possible to adjust the text encoding, separator, text identifier, decimal separator and date format.

These settings are saved and restored the next time OnyxQuery is used on this workstation.

## 4. Requests via HTTP

To be able to send queries to OnyxQuery via HTTP, OnyxQuery must be running on the computer and the HTTP server must be enabled.

### 4.1. Save results as a file

In OnyxQuery, the calls described so far can also be started via HTTP. The results are saved in files with the name passed.

#### Syntax

```
http://Server:Port/QueryFile <Parameter>
```

#### Beispiel

```
http://127.0.0.1:12000/QueryFile -ImgFile="D:\Res.obj" -QueryMode=F  
-ImgCode="100002VSS" -Option="OBJ"
```

### 4.2. Results received back in response to the request

If information is to be requested as XML or CSV, the string <Response> must be specified as the file name. In QueryMode **F** the requested image file is returned in the answer, a specification after -ImgFile is not necessary.

#### Syntax

```
http://Server:Port/QueryResponse <Parameter>
```

#### Beispiele

Verfügbare Analysen zum Tracingtypen 30 als XML erfregen:

```
http://MyServer:12000/QueryResponse -QueryMode=A -XML=<Response>  
-TracingTypeID=30
```

Ein 2D-Bild als JPG abfragen:

```
http://127.0.0.1:12000/QueryResponse -QueryMode=F -ImgCode=10000M2GE -Option=JPG
```

## 5. Hand over server connection at startup

As with the programs **OnyxCeph**, **OnyxStat**, **OnyxRegister** and **OnyxAdmin**, **OnyxQuery** can also be passed a database connection at startup. The program will then try to connect to the passed server first. After the parameter:

```
-ConnectMode=1
```

can be specified with which protocol to connect to the database server. Possible values here are:

```
"0": Search for the server  
"1": Connection via „TCP/IP“  
"2": Connection via "NamedPipes"
```

After the parameter:

```
-ConnectSName= "[MyServer]@MyServer"
```

the name of the server can be specified. This is composed of the name of the server computer in square brackets, followed by the "@" character and then a valid connection.

Gültige Servernamen sind zum Beispiel:

- `[OnyxServer]@OnyxServer`
- `[DBServer]@local.praxis.de`
- `[MyServer]@192.12.13.1`

If the OnyxDBServer serves a port other than 16300, this must be specified after the server name:

```
[OnyxServer]@OnyxServer:17300
```

## 6. Detect when request has been processed

If **OnyxQuery** is started with a query, e.g. in a batch file, the process will not return until the query has been processed or an error has occurred.

In order to detect in a calling programme when the processing is finished, there is also the possibility that OnyxQuery sends a Windows message:

`-ReadyMsg= MSG`

OnyxQuery sends the BROADCAST-Windows **MSG** after completion.

This must be registered with the operating system using RegisterWindowMessage.

If the query was processed successfully, wParam is 0, in case of error 1.

The number of written data records is returned in lParam.

The API call in **OnyxQuery** looks like this:

```
PostMessage (HWND_BROADCAST, RegisterWindowMessage(MSG), Error, RecordCount);
```

## 7. Parameter -Action

Via the parameter:

`-Action=???`

certain actions can be executed in OnyxQuery. Currently the following actions are supported:

`-Action=Hide` OnyxQuery is minimised

`-Action=Close` OnyxQuery will be closed after the execution of the current query

## 8. Overview of necessary parameters

Parameter	Patienten	Images	Tracingtypes	Analyses	Results	File
QueryMode	P	I	T	A	R	F
XML und/oder CSV	*	*	*	*	*	
ImgFile						*
Client	(*)	(*)			(*)	(*)
PatID			*		(*)	
ImageTypeID		(*)	*			
ImageTypeCode		(*)	*			
TracingTypeID				*	*	
AnaCode					*	
ImgCode					*	*
Option						(*)
ReadyMsg	(*)	(*)	(*)	(*)	(*)	(*)
Action	(*)	(*)	(*)	(*)	(*)	(*)

\* Please use the parameter ImageTypeID or ImageTypeCode.

## Appendix

**Table 1: Codes of the image types for the parameter ImageTypeCode**

<b>Image type</b>	<b>Image type (for Code)</b>	<b>Code</b>
Lateral Right	PhotoLateralRight	PLR
Frontal	PhotoFrontal	PF
Frontal Smile	PhotoFrontalSmile	PFS
Oblique Right	PhotoObliqueRight	POR
Oblique Left	PhotoObliqueLeft	POL
Lateral Left	PhotoLateralLeft	PLL
No Type	NoType 2D	NO2D
Oblique Open Right	PhotoObliqueRightOpen	PORO
Oblique Open Left	PhotoObliqueLeftOpen	POLO
Oblique Smile Right	PhotoObliqueRightSmile	PORS
Oblique Smile Left	PhotoObliqueLeftSmile	POLS
Frontal	PhotoFrontalOpen	PFO
Lateral Open Left	PhotoLateralLeftOpen	PLLO
Lateral Open Right	PhotoLateralRightOpen	PLRO
Lateral Smile Right	PhotoLateralRightSmile	PLRS
Lateral Smile Left	PhotoLateralLeftSmile	PLLS
Bottom View	FotoFrontalSMV	PFSMV
FaceScan	FaceScan3D	FS3D
Frontal Mouth Wide Open	PhotoFrontalMouthWideOpen	PFMWO
Frontal Mouth Relaxed	PhotoFrontalMouthRelaxed	PFMR
No Type 3D	NoType 3D	NO3D
Nose Mouth	PhotoFrontalNoseMouth	PFNM
Nose	FotoFrontalNose	PFN
Crown	PhotoCrown	PC
USER 1	USER1	US1
USER 2	USER2	US2
USER 3	USER3	US3
USER 4	USER4	US4
USER 5	USER5	US5
USER 6	USER6	US6
USER 7	USER7	US7
Lateral Ceph	XRyCephalogramRight	XRCR
PA Ceph	XRyCephalogramPostAnt	XRCPA
Panoramic	XRyPanoramic	XR PAN
SMV	XRySMV	XRS MV
Lateral Ceph (Left)	XRyCephalogramLeft	XRCL
AP Ceph	XrayCephalogramAntPost	XRCAP
Left Hand	XrayHandLeft	XRHL
Right Hand	XRyHandRight	XRHR
Teeth	XRyTeeth	XRTT
CT-HeadScan	CTHead3D	CTH3D
Tooth 1	XRyTooth1	XRT1
Tooth 2	XRyTooth2	XRT2
Tooth 3	XRyTooth3	XRT3
Tooth 4	XRyTooth4	XRT4
Tooth 5	XRyTooth5	XRT5
Tooth 6	XRyTooth6	XRT6
Tooth 7	XRyTooth7	XRT7
TMJ Right	XrayTMJRight	XTMJR
TMJ Left	XRyTMJLeft	XTMJL
TMJ	XRyTMJ	XRTMJ
Model Upper Occlusal	CAstMaxillaOcclusal	CAMAXO
Model Lower Occlusal	CAstMandibulaOcclusal	CAMANDO
Model Right Buccal	CAstBuccalRight	CABR
Model Anterior Dental	CAstAntDental	CAAD
Cast Left Buccal	CAstBuccalLeft	CABL
Cast Deciduous Dentition	CAstDeciduous	CAD

Cast Permanent Dentition	CAstPermanent	CAP
Cast Mixed Dentition	CAstMixed	CAM
Cast Permanent 3D	CAstPermanent3D	CAP3D
Cast Deciduous 3D	CAstDeciduous3D	CAD3D
Cast Mixed 3D	CAstMixed3D	CAM3D
Cast Maxilla 3D	CAstMaxilla3D	CAMAX3D
Cast Mandibula 3D	CAstMandibula3D	CAMAND3D
Overlay 3D	OVerLay3D	OV3D
Upper Occlusal	IntraOralMaxillaOcclusal	IOMAXO
Lower Occlusal	IntraOralMandibulaOcclusal	IOMANDO
Right Occlusion	IntraOralOcclusionRight	IOOR
Anterior Occlusion	IntraOralOcclusionAnterior	IOOA
Left Occlusion	IntraOralOcclusionLeft	IOOL
Overjet Left	IntraOralOverJetLeft	IOOJL
Overjet Right	IntraOralOverjetRight	IOOJR
Anterior Occlusion Relaxed	IntraOralOcclusionAnteriorRelaxed	IOOAR
Anterior Occlusion Wide Open	IntraOralOkklusionAnteriorWideOpen	IOOAWO



**image** *instruments*

© Image Instruments GmbH  
[info@image-instruments.de](mailto:info@image-instruments.de)  
[www.onyx-ceph.de](http://www.onyx-ceph.de)