**\*\*\*\*\*\*\*\*\*\*https://en.wikipedia.org/wiki/Common\_Object\_Request\_Broker\_Architecture**

The Common Object Request Broker Architecture (CORBA) is a standard defined by the Object Management Group (OMG) designed to facilitate the communication of systems that are deployed on diverse platforms. CORBA enables collaboration between systems on different operating systems, programming languages, and computing hardware. CORBA uses an object-oriented model although the systems that use the CORBA do not have to be object-oriented. CORBA is an example of the distributed object paradigm.

معماری کارگزار Common Object Request (CORBA) یک استاندارد تعریف شده توسط گروه مدیریت شی (OMG) است که برای تسهیل ارتباط سیستم هایی که در سیستم عامل های مختلف مستقر هستند طراحی شده است CORBA. همکاری بین سیستم ها در سیستم عامل های مختلف ، زبان های برنامه نویسی و سخت افزار محاسباتی را امکان پذیر می کند CORBA. از یک مدل شی گرا استفاده می کند اگرچه سیستم هایی که از CORBA استفاده می کنند لازم نیست شی گرا باشند. CORBA نمونه ای از الگوی شی توزیع شده است.

CORBA enables communication between software written in different languages and running on different computers. Implementation details from specific operating systems, programming languages, and hardware platforms are all removed from the responsibility of developers who use CORBA. CORBA normalizes the method-call semantics between application objects residing either in the same address-space (application) or in remote address-spaces (same host, or remote host on a network). Version 1.0 was released in October 1991.

CORBA ارتباط بین نرم افزارهایی را که به زبانهای مختلف نوشته شده و در رایانه های مختلف اجرا می شود ، امکان پذیر می کند. جزئیات پیاده سازی سیستم عامل های خاص ، زبان های برنامه نویسی و سیستم عامل های سخت افزاری همه از مسئولیت برنامه نویسان استفاده کننده از CORBA برداشته می شوندCORBA. معنای تماس متد را بین اشیاء برنامه که در همان مکان آدرس (برنامه) یا در مکانهای از راه دور (همان میزبان ، یا میزبان از راه دور در یک شبکه) ساکن هستند ، عادی می کند. نسخه 1.0 در اکتبر 1991 منتشر شد.

CORBA uses an interface definition language (IDL) to specify the interfaces that objects present to the outer world. CORBA then specifies a mapping from IDL to a specific implementation language like C++ or Java. Standard mappings exist for Ada, C, C++, C++11, COBOL, Java, Lisp, PL/I, Object Pascal, Python, Ruby and Smalltalk. Non-standard mappings exist for C#, Erlang, Perl, Tcl and Visual Basic implemented by object request brokers (ORBs) written for those languages.

CORBA از یک زبان تعریف رابط (IDL) برای مشخص کردن واسطهای موجود در جهان بیرونی استفاده می کند. سپس CORBA نقشه برداری از IDL به زبان اجرای خاص مانند C ++ یا Java را مشخص می کند. نگاشتهای استاندارد برای Ada، C، C ++، C ++ 11، COBOL، Java، Lisp، PL / I، Object Pascal، Python، Ruby و Smalltalk وجود دارد. نقشه برداری های غیر استاندارد برای C # ، Erlang ، Perl ، Tcl و Visual Basic وجود دارد که توسط کارگزاران درخواست شی (ORB) برای آن زبانها نوشته شده است.

The CORBA specification dictates there shall be an ORB through which an application would interact with other objects. This is how it is implemented in practice:

مشخصات CORBA بیان می کند که باید یک ORB وجود داشته باشد که از طریق آن یک برنامه با اشیاء دیگر ارتباط برقرار کند. این نحوه اجرا در عمل است:

\*The application simply initializes the ORB, and accesses an internal Object Adapter, which maintains things like reference counting, object (and reference) instantiation policies, and object lifetime policies.

\* برنامه به سادگی ORB را آغاز می کند ، و به یک آداپتور داخلی (Object Adapter) نیز دسترسی پیدا می کند ، که مواردی مانند شمارش مرجع ، خط مشی های مربوط به فوریت های مربوط به شمارش (و مرجع) و خط مشی های طول عمر را حفظ می کند.

\* The Object Adapter is used to register instances of the generated code classes. Generated code classes are the result of compiling the user IDL code, which translates the high-level interface definition into an OS- and language-specific class base for use by the user application. This step is necessary in order to enforce CORBA semantics and provide a clean user process for interfacing with the CORBA infrastructure.

\* از آداپتور Object برای ثبت موارد کلاسهای کد تولید شده استفاده می شود. کلاسهای ایجاد شده در نتیجه حاصل تدوین کد IDL کاربر است که تعریف رابط سطح بالا را به یک کلاس کلاس OS و زبان خاص برای استفاده توسط برنامه کاربر ترجمه می کند. این مرحله به منظور اجرای معانی شناسی CORBA و فراهم آوردن یک فرآیند کاربری پاک برای ارتباط با زیرساختهای CORBA ضروری است.

**\*\*\*\*\*\*\*\*\*\*https://en.wikipedia.org/wiki/Object\_request\_broker**

**Object request broker**

In distributed computing, an object request broker (ORB) is a middleware which allows program calls to be made from one computer to another via a computer network, providing location transparency through remote procedure calls. ORBs promote interoperability of distributed object systems, enabling such systems to be built by piecing together objects from different vendors, while different parts communicate with each other via the ORB.

در محاسبات توزیع شده ، یک کارگزار درخواست شی (ORB) واسطه ای است که امکان برقراری تماس های برنامه از یک رایانه به دیگری از طریق شبکه رایانه ای را فراهم می کند و شفاف سازی مکان را از طریق تماس های رویه ای از راه دور فراهم می کند. ORB ها قابلیت همکاری سیستم های اشیاء توزیع شده را ارتقا می بخشند و این امکان را می دهند که با جمع کردن اشیاء از فروشندگان مختلف ، چنین سیستمهایی ساخته شوند ، در حالی که قسمت های مختلف از طریق ORB با یکدیگر ارتباط برقرار می کنند.

ORBs handle the transformation of in-process data structures to and from the raw byte sequence, which is transmitted over the network. This is called marshalling or serialization. In addition to marshalling data, ORBs often expose many more features, such as distributed transactions, directory services or real-time scheduling. Some ORBs, such as CORBA-compliant systems, use an interface description language to describe the data that is to be transmitted on remote calls.[1][2]

ORB ها تبدیل ساختارهای داده داده در فرآیند را به و از توالی بایت خام که از طریق شبکه منتقل می شود ، انجام می دهند. به این حالت مارشال یا سریال سازی گفته می شود. علاوه بر داده های باتلاقی ، ORB ها اغلب ویژگی های بیشتری مانند معاملات توزیع شده ، خدمات دایرکتوری یا برنامه ریزی در زمان واقعی را در معرض دید شما قرار می دهند. برخی از ORB ها ، مانند سیستم های سازگار با CORBA ، از یک زبان توصیف رابط برای توصیف داده هایی که باید در تماس های از راه دور منتقل شوند استفاده می کنند. [1] [2]

In object-oriented languages, an ORB actually provides a framework which enables remote objects to be used over the network, in the same way as if they were local and part of the same process. On the client side, so-called stub objects are created and invoked, serving as the only part visible and used inside the client application. After the stub's methods are invoked, client-side ORB performs the marshalling of invocation data, and forwards the request to the server-side ORB. On the server side, ORB locates the targeted object, executes the requested operation, and returns the results. Having the results available, client's ORB performs the demarshalling and passes the results back into the invoked stub, making them available to the client application. The whole process is transparent, resulting in remote objects appearing as if they were local.[1]

در زبانهای شی گرا ، یک ORB در واقع چارچوبی را فراهم می کند که اشیاء از راه دور را بتوان از طریق شبکه استفاده کرد ، به همان روشی که گویا آنها محلی و بخشی از همان فرآیند هستند. در سمت مشتری ، اشیاء خرد خرد به وجود می آیند و مورد استناد قرار می گیرند ، به عنوان تنها بخشی که در برنامه مشتری مشاهده و استفاده می شود ، خدمت می کند. پس از فراخوانی روشهای خرد خرد ، ORB سمت مشتری باعث تکمیل داده های فراخوانی می شود و درخواست را به سمت سرور ORB ارسال می کند. در سمت سرور ، ORB هدف مورد نظر را پیدا می کند ، عملیات درخواست شده را انجام می دهد و نتایج را برمی گرداند. با در دسترس بودن نتایج ، ORB مشتری لیست بندی مجدد را انجام می دهد و نتایج را به خرد خرد شده منتقل می کند و آنها را در دسترس مشتری قرار می دهد. کل فرایند شفاف است ، در نتیجه اشیاء از راه دور به نظر می رسد که محلی هستند. [1]

**\*\*\*\*\*\*\*\*\*\*docs.oracle.com**

CORBA:

The Application Server supports a standard set of protocols and formats that ensure interoperability. Among these protocols are those defined by CORBA.

سرور برنامه از یک مجموعه استاندارد پروتکل و قالب پشتیبانی می کند که قابلیت همکاری را تضمین می کند. در بین این پروتکل ها مواردی که توسط CORBA تعریف شده است.

The CORBA (Common Object Request Broker Architecture) model is based on clients requesting services from distributed objects or servers through a well-defined interface by issuing requests to the objects in the form of remote method requests. A remote method request carries information about the operation that needs to be performed, including the object name (called an object reference) of the service provider and parameters, if any, for the invoked method. CORBA automatically handles network programming tasks such as object registration, object location, object activation, request de-multiplexing, error-handling, marshalling, and operation dispatching.

مدل CORBA (معماری کارگزار درخواست مشترک شیء) مبتنی بر مشتریانی است که از طریق یک رابط کاربری تعریف شده با صدور درخواست به اشیاء در قالب درخواست روش از راه دور ، خدمات را از اشیاء یا سرورهای توزیع شده درخواست می کنند. یک درخواست از راه دور دارای اطلاعاتی در مورد عملیاتی است که باید انجام شود ، از جمله نام شی (به نام مرجع شی) از ارائه دهنده خدمات و پارامترها ، در صورت وجود ، برای روش فراخوانده شده است. CORBA به طور خودکار وظایف برنامه نویسی شبکه از قبیل ثبت شی ، مکان یابی ، فعال سازی شیء ، درخواست تعدیل تعدیل ، کار با خطا ، باتلاق و اعزام عملیات را انجام می دهد.

What is the ORB?

The Object Request Broker (ORB) is the central component of CORBA. The ORB provides the required infrastructure to identify and locate objects, handle connection management, deliver data, and request communication.

کارگزار Object Request (ORB) جزء اصلی CORBA است. ORB زیرساخت های لازم را برای شناسایی و مکان یابی اشیاء ، رسیدگی به مدیریت اتصال ، تحویل داده ها و درخواست ارتباط برقرار می کند.

A CORBA object never talks directly with another. Instead, the object makes requests through a remote stub to the ORB running on the local machine. The local ORB then passes the request to an ORB on the other machine using the Internet Inter-Orb Protocol (IIOP for short). The remote ORB then locates the appropriate object, processes the request, and returns the results.

یک هدف CORBA هرگز به طور مستقیم با شخص دیگری صحبت نمی کند. در عوض ، شی از طریق یک خرد خرد از راه دور به ORB که در دستگاه محلی کار می کند ، درخواست می کند. سپس ORB محلی درخواست خود را با استفاده از پروتکل Inter-Orb Internet (کوتاه مدت IIOP) به دستگاه ORB منتقل می کند. سپس ORB از راه دور جسم مناسب را پیدا می کند ، درخواست را پردازش می کند و نتایج را برمی گرداند.

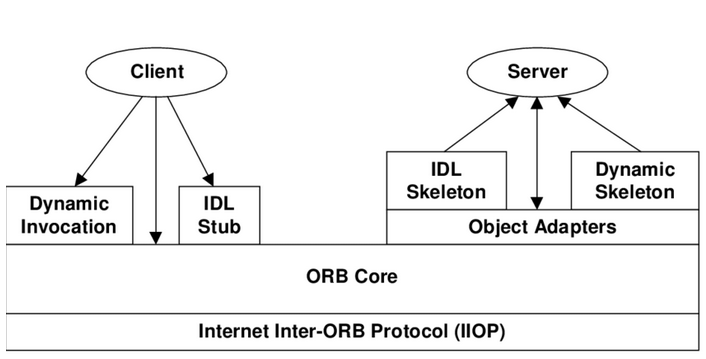
IIOP can be used as a Remote Method Invocation (RMI) protocol by applications or objects using RMI-IIOP. Remote clients of enterprise beans (EJB modules) communicate with the Application Server via RMI-IIOP.

IIOP توسط برنامه ها یا اشیاء با استفاده از RMI-IIOP می تواند به عنوان پروتکل Invocation Method Remote (RMI) از راه دور استفاده شود. مشتریان از راه دور لوبیای سازمانی (ماژول های EJB) از طریق RMI-IIOP با سرور برنامه ارتباط برقرار می کنند.

IIOP Listeners

An IIOP listener is a listen socket that accepts incoming connections from the remote clients of enterprise beans and from other CORBA-based clients. Multiple IIOP listeners can be configured for the Application Server. For each listener, specify a port number, a network address, and optionally, security attributes.

شنونده IIOP یک سوکت گوش است که اتصالات ورودی را از مشتری های از راه دور لوبیای سازمانی و سایر مشتری های مستقر در CORBA می پذیرد. شنوندگان متعدد IIOP را می توان برای سرور برنامه پیکربندی کرد. برای هر شنونده ، شماره پورت ، آدرس شبکه و ویژگی های امنیتی را به صورت اختیاری مشخص کنید.



**https://docs.oracle.com/javase/7/docs/api/org/omg/CORBA/ORB.html**

Short for Object Request Broker, a component in the CORBA programming model that acts as the middleware between clients and servers. In the CORBA model, a client can request a service without knowing anything about what servers are attached to the network. The various ORBs receive the requests, forward them to the appropriate servers, and then hand the results back to the client.

کوتاه برای Object Request Broker ، مؤلفه ای در مدل برنامه نویسی CORBA که به عنوان واسطه بین مشتری و سرور عمل می کند. در مدل CORBA ، مشتری می تواند از خدمات بدون اطلاع از سرورهای متصل به شبکه درخواست سرویس کند. ORB های مختلف درخواست ها را دریافت می کنند ، آنها را به سرورهای مناسب ارسال می کنند و سپس نتایج را به مشتری ارائه می دهند.

public abstract class ORB

extends Object

A class providing APIs for the CORBA Object Request Broker features. The ORB class also provides "pluggable ORB implementation" APIs that allow another vendor's ORB implementation to be used.

کلاس انتزاعی عمومی ORB  
شی را گسترش می دهد  
  
کلاس ارائه دهنده API ها برای ویژگی های کارگزار درخواست درخواست شی. کلاس ORB همچنین API های "ORB قابل اجرا" را امکان پذیر می سازد که امکان استفاده از اجرای ORB فروشنده دیگری را می دهد.

An ORB makes it possible for CORBA objects to communicate with each other by connecting objects making requests (clients) with objects servicing requests (servers).

ORB این امکان را می دهد تا اشیاء CORBA با اتصال اشیاء ایجاد درخواست (مشتری) با درخواست های سرویس دهنده (سرورها) با یکدیگر ارتباط برقرار کنند.

The ORB class, which encapsulates generic CORBA functionality, does the following: (Note that items 5 and 6, which include most of the methods in the class ORB, are typically used with the Dynamic Invocation Interface (DII) and the Dynamic Skeleton Interface (DSI). These interfaces may be used by a developer directly, but most commonly they are used by the ORB internally and are not seen by the general programmer.)

کلاس ORB که عملکردهای CORBA عمومی را محاصره می کند ، موارد زیر را انجام می دهد: (توجه داشته باشید که موارد 5 و 6 که شامل اکثر روش های موجود در کلاس ORB است ، به طور معمول با رابط پویایی Invocation (DII) و رابط اسکلت پویا (استفاده می شود) DSI). این رابط ها ممکن است توسط یک توسعه دهنده به طور مستقیم استفاده شود ، اما بیشتر آنها توسط ORB داخلی استفاده می شود و توسط برنامه نویس کلی دیده نمی شود.)

The ORB class can be used to obtain references to objects implemented anywhere on the network.

از کلاس ORB می توان برای به دست آوردن اشاراتی به اشیاء اجرا شده در هر نقطه از شبکه استفاده کرد.