IW 7535866

THE UNITED STATES OF AMERICA

TO ALL TO WHOM THESE PRESENTS SHALL COME:

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office

June 29, 2015

THIS IS TO CERTIFY THAT ANNEXED IS A TRUE COPY FROM THE RECORDS OF THIS OFFICE OF THE FILE WRAPPER AND CONTENTS OF:

APPLICATION NUMBER: *09/629,577*
FILING DATE: *July 31, 2000*
PATENT NUMBER: *6,732,147*
ISSUE DATE: *May 04, 2004*

By Authority of the
Under Secretary of Commerce for Intellectual Property
and Director of the United States Patent and Trademark Office

EXHIBIT
Ex. 1002
Volume 2

M. TARVER
Certifying Officer

PART (2) OF (6) PART(S)

- **object_key** contains opaque data used to identify the object that is the target of the operation[5]. Its value is obtained from the **object_key** field of the **TAG_INTERNET_IOP** profile or the **TAG_COMPLETE_OBJECT_KEY** component of the **TAG_MULTIPLE_COMPONENTS** profile.
- **operation** contains the name of the CORBA operation being invoked. The case of the operation name must match the case of the operation name specified in the OMG IDL source for the interface being used.

  Attribute accessors have names as follows:
  - Attribute selector: operation name is "_get_<attribute>"
  - Attribute mutator: operation name is "_set_<attribute>"

  CORBA::Object pseudo-operations have operation names as follows:
  - **get_interface** - operation name is "_interface"
  - **get_implementation** - operation name is "_implementation"
  - **is_a** - operation name is "_is_a"
  - **non_existent** - operation name is "_non_existent"
- **Principal** contains a value identifying the requesting principal. No particular meaning or semantics are associated with this value. It is provided to support the `BOA::get_principal` operation.

16.4.1.2 *Invoke request body*

The invoke request body contains the following items encoded in this order:

- All **in** and **inout** parameters, in the order in which they are specified in the operation's OMG IDL definition, from left to right.
- An optional Context pseudo object, encoded as described in Section 15.3.5.4, "Context," on page 15-29[6]. This item is only included if the operation's OMG IDL definition includes a context expression, and only includes context members as defined in that expression.

16.4.2 *DCE-CIOP Invoke Response Message*

Invoke response messages are returned from servers to clients as the `response_message` parameter of an `invoke` RPC.

5. Previous revisions of DCE-CIOP included an endpoint_id member, obtained from an optional TAG_ENDPOINT_ID component, as part of the object identity. The endpoint ID, if used, is now contained within the object key, and its position is specified by the optional TAG_ENDPOINT_ID_POSITION component.

6. Previous revisions of DCE-CIOP encoded the Context in the InvokeRequestHeader. It has been moved to the body for consistency with GIOP.

Like invoke request messages, an invoke response message is made up of a header and a body. The header has a fixed format, while the format of the body depends on the operation's OMG IDL definition and the outcome of the invocation.

16.4.2.1 *Invoke response header*

DCE-CIOP invoke response headers have the following structure:

```
module DCE_CIOP {                                    // IDL
    enum InvokeResponseStatus {
        INVOKE_NO_EXCEPTION,
        INVOKE_USER_EXCEPTION,
        INVOKE_SYSTEM_EXCEPTION,
        INVOKE_LOCATION_FORWARD,
        INVOKE_TRY_AGAIN
    };

    struct InvokeResponseHeader {
        boolean byte_order;
        IOP::ServiceContextList service_context;
        InvokeResponseStatus status;

        // if status = INVOKE_NO_EXCEPTION,
        // result then inouts and outs follow


        // if status = INVOKE_USER_EXCEPTION or
        // INVOKE_SYSTEM_EXCEPTION, an exception follows

        // if status = INVOKE_LOCATION_FORWARD, an
        // IOP::IOR follows
    };
};
```

The members have the following definitions:

- **byte_order** indicates the byte ordering used in the representation of the remainder of the message. A value of FALSE indicates big-endian byte ordering, and TRUE indicates little-endian byte ordering.
- **service_context** contains any ORB service data that needs to be sent from the client to the server.
- **status** indicates the completion status of the associated request, and also determines the contents of the body.

16.4.2.2 *Invoke Response Body*

The contents of the invoke response body depends on the value of the `status` member of the invoke response header, as well as the OMG IDL definition of the operation being invoked. Its format is one of the following:

- If the **status** value is **INVOKE_NO_EXCEPTION,** then the body contains the operation result value (if any), followed by all inout and out parameters, in the order in which they appear in the operation signature, from left to right.
- If the **status** value is **INVOKE_USER_EXCEPTION** or **INVOKE_SYSTEM_EXCEPTION**, then the body contains the exception, encoded as in GIOP.
- If the **status** value is **INVOKE_LOCATION_FORWARD**, then the body contains a new IOR containing a **TAG_INTERNET_IOP** or **TAG_MULTIPLE_COMPONENTS** profile whose components can be used to communicate with the object specified in the invoke request message[7]. This profile must provide at least one new DCE-CIOP binding component. The client ORB is responsible for resending the request to the server identified by the new profile. This operation should be transparent to the client program making the request. See "DCE-CIOP Object Location" on page 16-21 for more details.
- If the **status** value is **INVOKE_TRY_AGAIN**, then the body is empty and the client should reissue the `invoke` RPC, possibly after a short delay[8].

16.4.3 *DCE-CIOP Locate Request Message*

Locate request messages may be sent from a client to a server, as the `request_message` parameter of a `locate` RPC, to determine the following regarding a specified object reference:

- Whether the object reference is valid.
- Whether the current server is capable of directly receiving requests for the object reference.
- If not capable, to solicit an address to which requests for the object reference should be sent.

For details on the usage of the `locate` RPC, see Section 16.6, "DCE-CIOP Object Location," on page 16-21.

Locate request messages contain a fixed-format header, but no body.

16.4.3.1 *Locate Request Header*

DCE-CIOP locate request headers have the following format:

```
module DCE_CIOP {                    // IDL
    struct LocateRequestHeader {
        boolean byte_order;
```

7. Previous revisions of DCE-CIOP returned a MultipleComponentProfile structure. An IOR is now returned to allow either a TAG_INTERNET_IOP or a TAG_MULTIPLE_COMPONENTS profile to be used.

8. An exponential back-off algorithm is recommended, but not required.

```
            sequence <octet> object_key;
            string operation;

            // no body follows
        };
    };
```

The members have the following definitions:

- **byte_order** indicates the byte ordering used in the representation of the remainder of the message. A value of FALSE indicates big-endian byte ordering, and TRUE indicates little-endian byte ordering.
- **object_key** contains opaque data used to identify the object that is the target of the operation. Its value is obtained from the **object_key** field of the **TAG_INTERNET_IOP** profile or the **TAG_COMPLETE_OBJECT_KEY** component of the **TAG_MULTIPLE_COMPONENTS** profile.
- **operation** contains the name of the CORBA operation being invoked. It is encoded as in the invoke request header.

16.4.4 DCE-CIOP Locate Response Message

Locate response messages are sent from servers to clients as the `response_message` parameter of a `locate` RPC. They consist of a fixed-format header, and a body whose format depends on information in the header.

16.4.4.1 Locate Response Header

DCE-CIOP locate response headers have the following format:

```
module DCE_CIOP {                               // IDL
    enum LocateResponseStatus {
        LOCATE_UNKNOWN_OBJECT,
        LOCATE_OBJECT_HERE,
        LOCATE_LOCATION_FORWARD,
        LOCATE_TRY_AGAIN
    };
    struct LocateResponseHeader {
        boolean byte_order;
        LocateResponseStatus status;

        // if status = LOCATE_LOCATION_FORWARD, an
        // IOP::IOR follows
    };
};
```

The members have the following definitions:

- **byte_order** indicates the byte ordering used in the representation of the remainder of the message. A value of FALSE indicates big-endian byte ordering, and TRUE indicates little-endian byte ordering.

DOCKET ALARM

Explore Litigation Insights

Docket Alarm provides insights to develop a more informed litigation strategy and the peace of mind of knowing you're on top of things.

Real-Time Litigation Alerts

Keep your litigation team up-to-date with **real-time alerts** and advanced team management tools built for the enterprise, all while greatly reducing PACER spend.

Our comprehensive service means we can handle Federal, State, and Administrative courts across the country.

Advanced Docket Research

With over 230 million records, Docket Alarm's cloud-native docket research platform finds what other services can't. Coverage includes Federal, State, plus PTAB, TTAB, ITC and NLRB decisions, all in one place.

Identify arguments that have been successful in the past with full text, pinpoint searching. Link to case law cited within any court document via Fastcase.

Analytics At Your Fingertips

Learn what happened the last time a particular judge, opposing counsel or company faced cases similar to yours.

Advanced out-of-the-box PTAB and TTAB analytics are always at your fingertips.

API

Docket Alarm offers a powerful API (application programming interface) to developers that want to integrate case filings into their apps.

LAW FIRMS
Build custom dashboards for your attorneys and clients with live data direct from the court.

Automate many repetitive legal tasks like conflict checks, document management, and marketing.

FINANCIAL INSTITUTIONS
Litigation and bankruptcy checks for companies and debtors.

E-DISCOVERY AND LEGAL VENDORS
Sync your system to PACER to automate legal marketing.

WHAT WILL YOU BUILD? | **sales@docketalarm.com** | **1-866-77-FASTCASE**

fastcase® Smarter legal research.