

Network Working Group
Internet-Draft
Intended status: Standards Track
Expires: March 11, 2011

M. Boucadair
France Telecom
E. Burgey
Orange Labs
September 7, 2010

A64: DNS Resource Record for IPv4-Embedded IPv6 Address
draft-boucadair-behave-dns-a64-02

Abstract

In the context of IPv4-IPv6 interconnection (or interworking), several solutions have been proposed within IETF. Some of these solutions require the definition of a new IPv4-Embedded IPv6 address which is used to represent an IPv4-only host in an IPv6 realms and the definition of a new mechanism called DNS64 for synthesizing a AAAA record, representing an IPv4-Embedded IPv6 address in the DNS system, from the A record representing the IPv4 address of the IPv4-only host . This IPv4-Embedded IPv6 address, when managed by a translator, is to be considered as "fake" address in a DNS system since it is not necessarily assigned to any host's interface qualified by the associated name.

This document defines a new DNS record type and query type to identify IPv4-Embedded IPv6 address from native IPv6 ones. The new record type and query type aim at helping the requesting party to enforce its policies and avoid crossing NAT64 boxes when possible. Native addresses are to be preferred.

Requirements Language

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in [RFC 2119](#) [[RFC2119](#)].

Status of this Memo

This Internet-Draft is submitted in full conformance with the provisions of [BCP 78](#) and [BCP 79](#).

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at <http://datatracker.ietf.org/drafts/current/>.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any

Internet-Draft

A64 DNS Record

September 2010

time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on March 11, 2011.

Copyright Notice

Copyright (c) 2010 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to [BCP 78](http://trustee.ietf.org/bcp78) and the IETF Trust's Legal Provisions Relating to IETF Documents (<http://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

This document may contain material from IETF Documents or IETF Contributions published or made publicly available before November 10, 2008. The person(s) controlling the copyright in some of this material may not have granted the IETF Trust the right to allow modifications of such material outside the IETF Standards Process. Without obtaining an adequate license from the person(s) controlling the copyright in such materials, this document may not be modified outside the IETF Standards Process, and derivative works of it may not be created outside the IETF Standards Process, except to format it for publication as an RFC or to translate it into languages other than English.

Internet-Draft

A64 DNS Record

September 2010

Table of Contents

1.	Introduction	4
1.1.	Overall Context	4
1.2.	Contribution of this Draft	4
1.3.	Backward Compatibility Issue	5
2.	Terminology	6
3.	Q/A	7
3.1.	Why a new RRTYPE is Required?	7
3.2.	Do We Need to Update all DNS Servers to Support this New RRTYPE?	8
3.3.	Does the Proposal Need Hosts to be Upgraded All?	9
3.4.	What is the Impact on the DNS Server Load?	10
3.5.	Are legacy Hosts Impacted?	10
3.6.	Location of Recursors	11
3.7.	How an A64-compliant Host Can Discover an A64-DNS Server?	11
4.	IPv4-Embedded IPv6 Record: A64	12
4.1.	A64 Record Type	12
4.2.	A64 Data Format	12
4.3.	A64 Query	12
4.4.	Textual format of A64 records	12
4.5.	IP64.INT Domain	12
4.6.	Modifications to Existing Query Types	12
4.7.	On the Need of a New Query type	13
5.	IANA Considerations	13
6.	Security Considerations	13
7.	Acknowledgements	13
8.	References	14
8.1.	Normative References	14
8.2.	Informative Reference	14
	Authors' Addresses	15

Internet-Draft

A64 DNS Record

September 2010

1. Introduction

1.1. Overall Context

Due to IPv4 address depletion, several solutions have been proposed within IETF. Among those solutions, IPv6 is the only perennial solution that should be implemented by service providers. Nevertheless, this implementation won't be in one shot and the co-existence between IPv4 and IPv6 should be managed for a while. In addition to the co-existence issue, interconnection means to enable successful communications between IPv4 and IPv6 realms must be enforced.

In this context, BEHAVE WG is currently specifying translator mechanisms which cover both stateful [[I-D.ietf-behave-v6v4-xlate-stateful](#)] and stateless [[I-D.ietf-behave-v6v4-xlate](#)] modes. The proposed solutions require the definition of a new IPv4-Embedded IPv6 address [[I-D.ietf-behave-address-format](#)] which is used to represent an IPv4-only host in an IPv6 realm. This type of addresses, when managed by a translator, is to be considered as "fake" address since it is not assigned to any host and can be confusing for the application. The application querying for this address must be aware that the application running on the host represented by the IPv4-Embedded IPv6 address is connected through an IPv4 interface and may be not able to use a reference (e.g., [[I-D.carpenter-behave-referral-object](#)]) to an IPv6 address in the packet payload.

When an IPv4-Embedded IPv6 address is used as destination address,

the traffic will be handled by a translator device and no direct communication would be possible.

[1.2.](#) Contribution of this Draft

The aforementioned "fake" addresses should not be confused with native IPv6 addresses in order to ease the enforcement of means to avoid translators whenever possible. Notifying to the requesting host that the resolved address is not a native address but an IPv4-Embedded IPv6 address (which is behind a NAT) would ease the local policies to prefer direct communications (i.e., avoid using IPv4-mapped IPv6 addresses when a native IPv6 address or a native IPv4 address is available).

For the sake of maintaining the reliability of the information maintained and sent by DNS and helping a given host to enforce policies to prefer native communications, a new record type is proposed (A64). This record type stores only IPv4-Embedded IPv6 addresses. A new query type A64 is also defined to request A64

records or synthesised A64 records.

This document does not make any assumption about the use of WKP or LIR IPv6 prefix to build IPv4-Embedded IPv6 addresses. Both schemes are supported.

[1.3.](#) Backward Compatibility Issue

The A64 record proposal suffers from DNS backward compatibility issue. This issue is to be positioned in the following context:

- o We are still in the early stages of IPv6 deployment and solutions to help smooth migration and ease the transition to full IPv6 should be encouraged;
- o IPv4-Embedded IPv6 addresses will be used in various architecture configurations and guidelines for the use of these addresses to populate DNS servers should be elaborated. This document is a contribution to this effort;
- o DNS64 proposal as defined in [[I-D.ietf-behave-dns64](#)] may be used for serving dual-stack hosts (e.g., context of multi-homing. The

device may be attached to distinct domains managed by distinct administrative entities. Means to ease the merging and the consistency of the received information should be implemented, see MIF WG problem statement). DNS servers do not have any means to restrict the use of the service based on the requesting connectivity capabilities. Appropriate solutions are to be put at disposal of the client;

- o [\[I-D.ietf-behave-dns64\]](#) does not define how to configure static records using IPv4-Embedded IPv6 addresses;
- o Not all DNS servers are to be upgraded. Only a subset of DNS servers may be updated to support the A64 records.
- o A procedure for the discovery of these A64-capable DNS servers is described in [\[I-D.boucadair-behave-dns64-discovery\]](#). When supported, only the appropriate DNS servers will be invoked.
- o In some contexts, referral procedure should be aware about the type of the remote party (IPv4-only applications running on top of an IPv6-only connected device). This may help avoiding "some" ALGs in the path.

[2.](#) Terminology

This document makes use of the following terms:

- o **Authoritative server:** is a DNS server that answers authoritatively for a given DNS query.
- o **Stub resolver:** denotes a resolver with minimum functionality, typically for use in end points that depend on a recursive resolver.
- o **Recursive resolver:** refers to a DNS server that accepts queries from a resolver, and asks another resolver for the answer on behalf of the first resolver.

- o Synthetic resource record (RR): denotes a DNS resource record that is not contained in any zone data file, but has been synthesized from other RRs. An example of synthetic RR is a AAAA record created from an A record.
- o DNS64 function: refers to a function that is provisioned with one or more IPv6 prefix(es) and is able to convert A64 records into AAAA records if there is no AAAA native records and also able to synthesize AAAA records from A records and the specific IPv6 prefix if there is no AAAA native records and no A64 records for the requested QNAME. As explained in [appendix B](#) of [\[I-D.ietf-behave-dns64\]](#) in some specific cases a DNS64 function may generate a synthetic AAAA records even if a native AAAA records still exists.
- o DNS_A64 function: A logical function that is provisioned with one or more IPv6 prefix(es) and is able to synthesize A64 records from A records and the specific prefix if there is no A64 records for the requested QNAME. As explained in [appendix B](#) of [\[I-D.ietf-behave-dns64\]](#) in some specific case a DNS_A64 function may generate a synthetic A64 records even if a native A64 records (generated by an authoritative server) still exists.
- o sDNS64 function : A simplified logical function that is able to convert A64 records into AAAA records if there is no AAAA native records. Such simplified function does not need to be provisioned with a specific IPv6 prefix.
- o Legacy DNS server : A DNS server that has not been updated to manage A64 records and that does not include DNS64 function or DNS_A64 function.

- o A64_DNS server : A DNS server that is able to store A64 records and process A64 queries. An A64_DNS does not include any DNS_A64 function or any DNS64 function
- o A64_DNS64 server: A server that provides all functionalities of a standard DNS server and includes also a DNS_A64 function.
- o DNS64 server: A server that provides all functionalities of a

standard DNS server and includes also a DNS64 function.

- o A64_DNS64 recursor: A recursive resolver that provides all functionalities of a standard DNS recursor and also provides the DNS_A64 function as part of its operation.
- o sDNS stub resolver : A stub resolver embedding a sDNS64 function which is able to manage A64 records and to issue A64 queries.

3. Q/A

3.1. Why a new RRTYPE is Required?

A new record type to enclose IPv4-Embedded IPv6 addresses is required for the following reasons:

1. DNS64 proposal as defined in [[I-D.ietf-behave-dns64](#)] is tailored for IPv6 only hosts. But it may be confusing for dual-stack hosts receiving an A records and a AAAA records for the same QNAME. If a native IPv6 address is enclosed in the DNS response, IPv6 communication should be selected. If the AAAA record contains an IPv4-Embedded IPv6 address, the dual-stack host should prefer direct IPv4 communications to avoid problems that could appear due to application layer gateway translation in the NAT64. Without an additional mechanism providing information of the nature of the address in the AAAA records or without a mechanism to restrict the use of the DNS64 service based on the requesting connectivity capabilities, dual-stack hosts won't be able to take appropriate decisions. Using A64 records for IPv4-Embedded IPv6 address is a way to distinguish them from native IPv6 address using AAAA records;
2. [[I-D.ietf-behave-dns64](#)] points out in [appendix A.4](#) the needs to introduce IPv4-Embedded IPv6 address in DNS server that are authoritative for a specific QNAME. The IPv6 prefix used to build this IPv4-Embedded IPv6 address has been chosen by the administrator of the requested QNAME and is not necessarily known by the operator providing the Internet access service to the requesting host. So to solve this problem, a mechanism is needed

to exchange information between Internet access provider to

distinguish an IPv4-Embedded IPv6 address from a native IPv6 address. Exchanging A64 records for IPv4-Embedded IPv6 address instead of AAAA records between DNS recursors and authoritative server solves this issue;

3. [[I-D.ietf-behave-dns64](#)] describes in [appendix B](#) cases where both IPv4-Embedded IPv6 address and native IPv6 can be returned in AAAA records and points out that without a specific mechanism to distinguish between them the standard process may choose the IPv4-Embedded IPv6 address. Using A64 records for IPv4-Embedded IPv6 address solves this issue;
4. A dual-stack application receiving an IPv4-Embedded IPv6 address is not aware that the destination application entity is using an IPv4 address. That application entity may not be able to understand any reference in the packet payload to an IPv6 address. With the introduction of A64 records, it is possible to develop a new API so that updated (i.e., A64-compliant) dual-stack applications are able to learn that the IPv6 address comes from an A64 records and is an IPv4-Embedded IPv6 address. Operating systems with this new API may embed a sDN64 function to deal with non updated IPv6 applications (i.e., non A64 compliant applications).

It was tempting to define only a new query type to indicated whether an enclosed AAAA record does not carry a native IPv6 address, but this does not allow configuring static entries in the DNS pointing to IPv4-Embedded IPv6 addresses.

[3.2.](#) Do We Need to Update all DNS Servers to Support this New RRType?

No.

A64_DNS servers or A64_DNS64 recursors are fully compatible with legacy DNS servers, recursors or stub resolvers because the answer to AAAA or to A queries and the recording or caching of AAAA records and A records in not modified. So they can be introduced in the existing DNS system without any trouble.

Owing to the DNS64 discovery mechanism described in [[I-D.boucadair-behave-dns64-discovery](#)], DNS64 recursors or A64_DNS64 recursor are able to distinguish A64-capable DNS servers from legacy DNS servers.

A64 resolvers, A64_DNS64 recursors or DNS64 recursors can issue non recursive requests to other servers to retrieve the address of an authoritative server for the QNAME and then use the A64 and DNS64

discovery mechanism to check if this authoritative server is an A64_DNS server. If the answer is that there is no A64_DNS server in this domain, this means that there is no A64 records for this QNAME, if the answer is the address of another A64_DNS server that is also authoritative for that domain, the resolver will send this request to this new server. With such a mechanism only one authoritative DNS server for this domain including IPv4-Embedded IPv6 must be upgraded toward to be A64-capable DNS. All intermediate DNS servers may be legacy DNS servers.

Only authoritative DNS servers recording an IPv4-Embedded IPv6 address must be updated to be A64-capable DNS servers but they don't need DNS64 or A64DN64 function. The IPv4-Embedded IPv6 address can be provisioned by standard processing. A DNS_A64 function may just be useful to simplify the provisioning process.

As in the [[I-D.ietf-behave-dns64](#)] proposal, only the DNS recursor serving IPv6-only host using the NAT64 service needs to be upgraded toward DNS64 recursor or A64_DNS64 recursor.

Other DNS recursor or DNS server can be legacy DNS server.

[3.3.](#) Does the Proposal Need Hosts to be Upgraded All?

No.

The DNS64 function is introduced to ensure compatibility with hosts or CPEs that are not updated. In a first step, to use the NAT64 service, it is not necessary to upgrade existing hosts or existing CPEs. It is sufficient to add a DNS64 recursor without modifying the legacy DNS recursor. During the DNS address provisioning phase (e.g., using DHCP or PPP), the Internet access service provider provisions the (one or a list of) IP address of a DNS recursor according to the connectivity option subscribed by the customer (i.e., IPv4 only, dual-stack or IPv6 only). For IPv4 only or dual-stack customers, the legacy DNS recursor or an A64_DNS64 recursor is recommended. For IPv6 only customer a DNS64 recursor is recommended.

During this first step, dual-stack hosts will never received IPv4-Embedded IPv6 address even if they are recorded in an authoritative A64 DNS server. IPv6 only hosts will have access to all NAT64. Some remaining issues of this first step are:

1. IPv6 applications running on IPv6 only host are not able to recognize that the application running on the other host may be IPv4 only and they may experience problem if they exchange

reference to an IPv6 address.

2. IPv4 only applications running on dual-stack host are not able to access to NAT64 service.

To solve the remaining issues, dual-stack hosts and CPEs must be upgraded with sDNS stub resolver and the legacy DNS recursor provided by the Internet access provider must be updated to an A64_DNS64 recursor. In the meantime, applications that want to distinguish IPv4-Embedded IPv6 addresses from native IPv6 ones must be upgraded to use a specific API to request A64 records.

The A64 and DNS64 discovery mechanism [[I-D.boucadair-behave-dns64-discovery](#)] can be used by dual-stack upgraded hosts to modify the pre-configured DNS recursor address with a A64_DNS64 recursor address.

As a result, upgraded hosts will be able to use the A64 and DNS64 discovery mechanism to modify their configuration if the DNS recursor address, provided by the Internet service access provider, is not the most appropriate one.

3.4. What is the Impact on the DNS Server Load?

The impact on the DNS server load would be low.

There is no need to store, in DNS system, information about the addressing capacity of hosts and the request/answer process remain fully stateless.

The amount of data to be stored in A64_DNS server, A64_DNS64 recursor or DNS64 recursor may increase if, for the same QNAME, both A64 records and AAAA records are present. But, this situation would be scarce.

Compared to DNS64 proposal [[I-D.ietf-behave-dns64](#)], a DNS64 recursor or an A64_DNS64 recursor can send three requests instead of two generating therefore extra processing in the DNS server, in the DNS64 recursor and in the A64_DNS64 recursor. But this extra processing can be avoided if a new request asking for AAAA, A64 and A records is specified (see [[I-D.li-dnsexst-ipv4-ipv6](#)]).

[3.5.](#) Are legacy Hosts Impacted?

No.

Legacy hosts are not impacted because the AAAA and the A records processing is the same in A64_DNS64 recursors and/or in A64_DNS servers as in legacy DNS servers.

If dual-stack or IPv4 only legacy hosts are provisioned with the address of a legacy DNS recursor or of an A64_DNS64 recursor, as recommended in this document, they will not be impacted.

If IPv6 only legacy hosts are provisioned with the address of a legacy DNS recursor or of an A64_DNS64 recursor, they will not be impacted.

If IPv6 only legacy hosts are provisioned with the address of a DNS64 recursor they will benefit from the NAT64 service.

[3.6.](#) Location of Recursors

Specific DNS64 and A64_DNS64 recursors are introduced in this document.

A DNS64 function must not be implemented in an authoritative server (for QNAME this server is authoritative for).

A DNS_A64 function must not be implemented in an authoritative server (for the QNAME this server is authoritative for) except for provisioning new A64 records in the DNS server.

The DNS64 recursor and the A64_DNS64 recursor must be the first DNS recursor managed by the operator providing the NAT64 service requested by the customer. The request coming from the stub-resolver of the host can pass only through recursors that are managed locally like a DNS recursor in a CPE before being intercepted by a DNS64 recursor or the A64_DNS64 recursor.

The DNS64 recursor or the A64_DNS64 recursor should not accept requests coming from other recursors managed by the operator or from

recursors managed by another Internet service provider. This is generally achieved by providing the address of the DNS64 recursor or A64_DNS64 recursor only to hosts or CPEs managed by customers of the Internet service provider. The address of an A64 DNS server (it may be an A64 DNS recursor) or a legacy DNS server may be provided to all other DNS servers, recursors or resolvers.

[3.7.](#) How an A64-compliant Host Can Discover an A64-DNS Server?

By default an host is provisioned by the Internet service provider with the address of a recursor. But an A64-compliant host can change this allocation for a most appropriate one thanks to the mechanism described in [[I-D.boucadair-behave-dns64-discovery](#)].

[4.](#) IPv4-Embedded IPv6 Record: A64

[[I-D.ietf-behave-address-format](#)] discusses issues related to how an IPv4 host can be represented in the IPv6 realm. A new address format is proposed. To store an IPv4-Embedded IPv6 address [[I-D.ietf-behave-address-format](#)], a new record type is defined.

[4.1.](#) A64 Record Type

The A64 resource record type is a record specific to the Internet class that is destined to store a single IPv4-Embedded IPv6 address.

A type value is to be assigned by IANA.

[4.2.](#) A64 Data Format

Same data format for as AAAA records [[RFC3596](#)].

[4.3.](#) A64 Query

An A64 query for a specified domain name in the Internet class returns all associated A64 resource records in the answer section of a response. As AAAA query type, a type A64 query does not perform additional section processing.

[4.4.](#) Textual format of A64 records

Same as for AAAA records [[RFC3596](#)].

[4.5.](#) IP64.INT Domain

In order to not pollute IP6.INT domain with IPv4-inferred records, a new domain SHOULD be defined.

The same requirements as those of IP6.INT are to be taken into account for IP64.INT.

[Editor note: input on the need of a dedicated domain or use IP6.INT would be welcomed. Authors are in favour of separating both domains in order to ease migration to IPv6-only Internet and avoid interference which may be induced by transition mechanisms.]

[4.6.](#) Modifications to Existing Query Types

All existing query types that perform type A and AAAA processing SHOULD be updated to perform type A, AAAA and A64 processing.

Only AAAA records SHOULD be returned when AAAA only type is specified

in the query.

But during a transition phase, some hosts with IPv6-only connectivity may not be updated to perform A64 queries. Specific IPv6 DNS servers, called DNS64 recursors, may be used to serve those hosts. A DNS64 recursor may return an A64 record as a response to AAAA query if no AAAA record is available for the specified resource (A64 records are converted to AAAA records in the answer). This behaviour advocates for enhancing the chance for a communication to be established even through a translator). DNS64 recursors SHOULD not be used by dual-stack hosts which do not support A64 query type. When A64 query type is received, only A64 record type MUST be returned.

[4.7.](#) On the Need of a New Query type

In the context of IPv4-IPv6 co-existence, new query type(s) would be required to achieve the following goals:

- o Retrieve both A64 and AAAA records by issuing one single query;
- o Retrieve both IPv4 and IPv6 records (i.e., A, A64 and AAAA) records by issuing one single query;

A candidate solution is described in [[I-D.li-dnsext-ipv4-ipv6](#)].

[5.](#) IANA Considerations

A record TYPE is to be assigned by IANA.

[6.](#) Security Considerations

This document does not introduce any security threat to the DNS system.

[7.](#) Acknowledgements

Authors would like to thank Andrew Sullivan for his constructive comments.

[8.](#) References

[8.1.](#) Normative References

- [RFC1034] Mockapetris, P., "Domain names - concepts and facilities", STD 13, [RFC 1034](#), November 1987.
- [RFC1035] Mockapetris, P., "Domain names - implementation and specification", STD 13, [RFC 1035](#), November 1987.
- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", [BCP 14](#), [RFC 2119](#), March 1997.

[RFC3596] Thomson, S., Huitema, C., Ksinant, V., and M. Souissi, "DNS Extensions to Support IP Version 6", [RFC 3596](#), October 2003.

[8.2.](#) Informative Reference

- [I-D.boucadair-behave-dns64-discovery]
Boucadair, M. and E. Burgey, "DNS64 Service Location and Discovery", [draft-boucadair-behave-dns64-discovery-00](#)", October 2009.
- [I-D.carpenter-behave-referral-object]
Carpenter, B., Boucadair, M., Halpern, J., Jiang, S., and K. Moore, "A Generic Referral Object for Internet Entities", [draft-carpenter-behave-referral-object-01](#) (work in progress), October 2009.
- [I-D.ietf-behave-address-format]
Bao, C., Huitema, C., Bagnulo, M., Boucadair, M., and X. Li, "IPv6 Addressing of IPv4/IPv6 Translators", [draft-ietf-behave-address-format-10](#) (work in progress), August 2010.
- [I-D.ietf-behave-dns64]
Bagnulo, M., Sullivan, A., Matthews, P., and I. Beijnum, "DNS64: DNS extensions for Network Address Translation from IPv6 Clients to IPv4 Servers", [draft-ietf-behave-dns64-10](#) (work in progress), July 2010.
- [I-D.ietf-behave-v6v4-xlate]
Li, X., Bao, C., and F. Baker, "IP/ICMP Translation Algorithm", [draft-ietf-behave-v6v4-xlate-22](#) (work in progress), August 2010.
- [I-D.ietf-behave-v6v4-xlate-stateful]
Bagnulo, M., Matthews, P., and I. Beijnum, "Stateful NAT64: Network Address and Protocol Translation from IPv6

Clients to IPv4 Servers",
[draft-ietf-behave-v6v4-xlate-stateful-12](#) (work in progress), July 2010.

[I-D.li-dnsexp-ipv4-ipv6]

Li, L., Li, Z., and X. Duan, "DNS Extensions to Support IPv4 and IPv6", [draft-li-dnsexp-ipv4-ipv6-02](#) (work in progress), October 2009.

Authors' Addresses

Mohamed Boucadair
France Telecom
3, Av Francois Chateau
Rennes, 35000
France

Email: mohamed.boucadair@orange-ftgroup.com

Eric Burgey
Orange Labs
38-40, rue du general Leclerc
Issy-les-Moulineaux Cedex 9 92794
France

Email: eric.burgey@orange-ftgroup.com