

# WebOTX V6

## JNDI のトラブルシューティング

改版履歴

2007 年 1 月 初版

## 目次

1	はじめに .....	1
2	JNDIサーバとの通信の仕組み .....	1
3	レプリケーションの動作 .....	2
4	JNDIサーバのレプリケーションの設定方法 .....	4
5	JNDIサーバとの通信障害の対処方法について .....	5
5.1.	Object Brokerサービスとの通信に失敗している場合 .....	5
5.2.	JNDIサーバとの通信に失敗している場合 .....	6

## 1 はじめに

本資料では、WebOTX で JNDI を使用するときが発生する障害の中から、お問い合わせの多い JNDI のレプリケーションに関する設定方法と、JNDI サーバとの通信障害への対処方法を説明しています。JNDI のレプリケーションは通常 Web アプリケーションの HTTP セッション情報をレプリケーションするために使用されます。

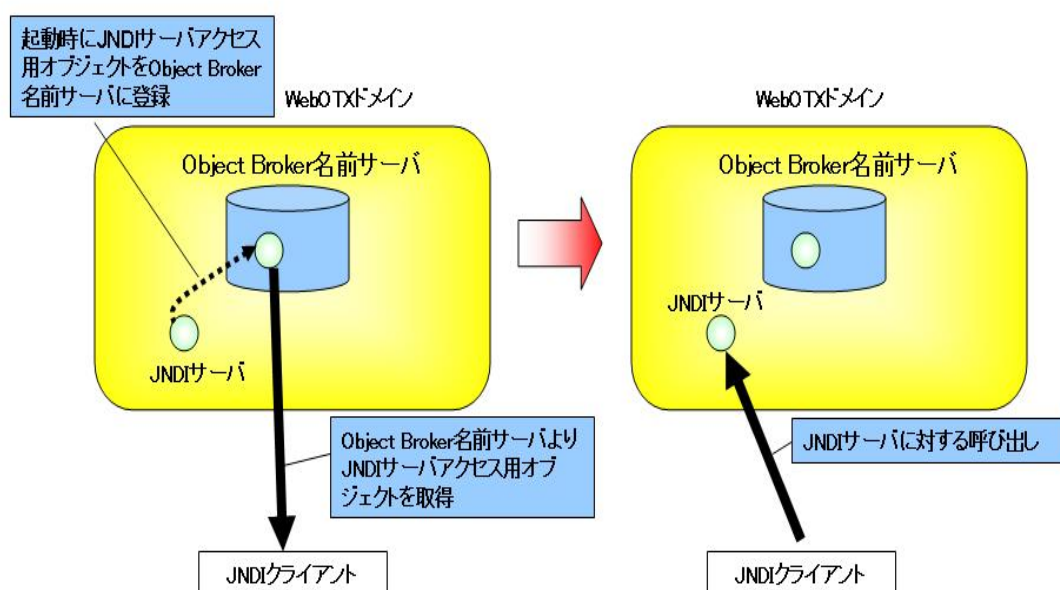
なお本資料は WebOTX Ver6 を対象としています。ただし Ver5 以前でも基本的な動作は同じですが設定方法は異なります。(Ver5 以前では JNDI サーバは単体で動作する Java サービスとなります)

## 2 JNDI サーバとの通信の仕組み

WebOTX JNDI の設定方法、障害対処方法について説明する前に、まず JNDI サーバとの通信を行うときの内部動作について説明します。

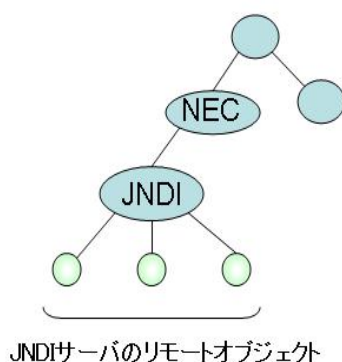
JNDI サーバ自身は RMI-IIOP プロトコルでクライアントからのリモートメソッド呼び出しを実行する Java のサーバです。これは WebOTX の JMX エージェントプロセスと同一の Java VM 上で動作します。

通常 Java の RMI 呼び出しを行う場合、RMI レジストリからサーバのリモートオブジェクトを取得して呼び出しますが、JNDI サーバの場合は WebOTX Object Broker の名前サーバから JNDI サーバにアクセスするためのリモートオブジェクトを取得して呼び出します。このオブジェクトは WebOTX ドメイン起動時に自ドメインの名前サーバに登録されます。



Object Broker の名前サーバからリモートオブジェクトを取得するときは、通常はオブジェクトにつけられた名前を指定して取得しますが、WebOTX JNDI の場合は少し異なります。登録されている名前を指定するのではなく、特定の名前コンテキスト下に登録されている全てのオブジェクトを取得して JNDI サーバとしてアクセスします。オブジェクトの名前は特に意識しません。これは複数の JNDI サーバで同じ名前情報を共有させるためにこのような処理になっています。特定のコンテキストとは実際には“NEC/JNDI”という名前のコンテキストになります。このコンテキストの下に既定値では“ホスト名:ドメイン名”という名前でリモートオブジェクトが登録されます。登録する名前は JNDI サーバの識別名の設定で変更可能ですが通常は変更する必要はありません。これはクラスタ環境で複数のサーバに同じ仮想ホスト名が割り当てられるような環境において、それぞれの JNDI サーバのリモートオブジェクトが同じ名前になって上書き登録されてしまうことを防ぐときに変更します。

Object Broker名前サーバの名前空間



JNDI のクライアント側から参照する Object Broker の名前サーバのロケーションを指定するには JNDI の `java.naming.provider.url` プロパティを指定します。指定形式は以下のとおりです。

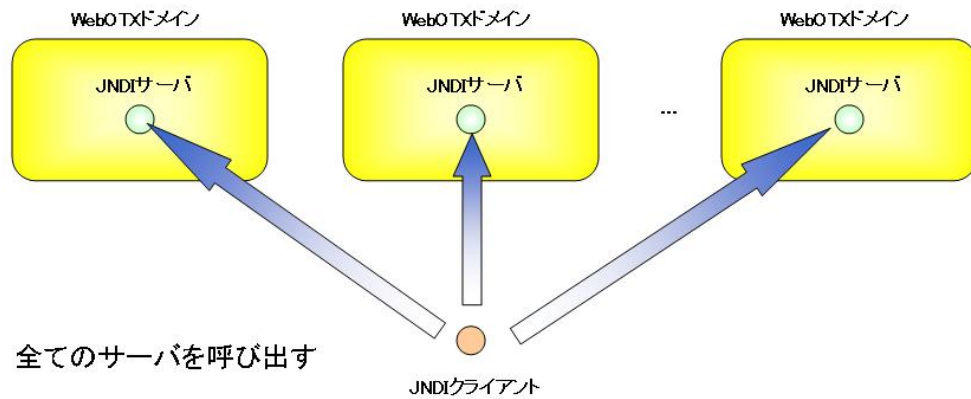
**corbaname://<ホスト名>:<名前サーバポート番号>**

### 3 レプリケーションの動作

複数の JNDI サーバがレプリケーションした場合の動作内容について説明します。Object Broker の名前サーバの JNDI 格納用コンテキスト(NEC/JNDI)下に複数の JNDI サーバのオブジェクトが登録されている場合は対応する JNDI サーバをレプリケーションさせるよう動作します。具体的な動作としては以下ようになります。

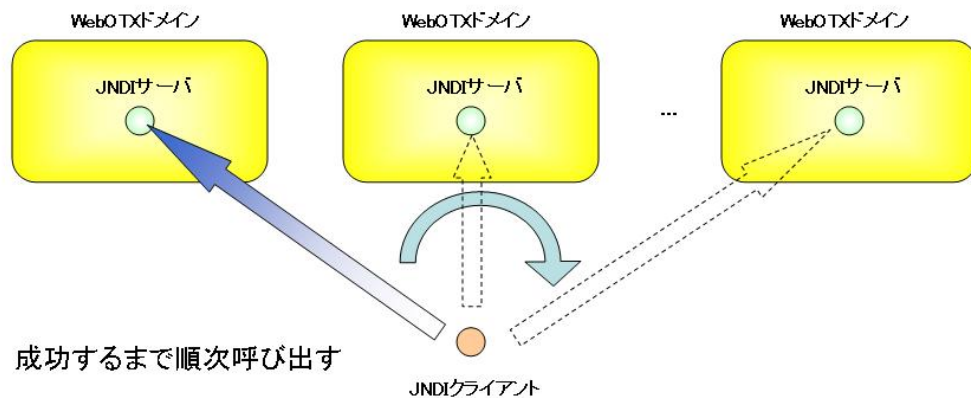
1. JNDI サーバへのオブジェクトを登録、削除は全てのサーバに対して呼び出す。

#### 名前の登録、削除



2. JNDI サーバからのオブジェクトの取得は、どれか一つのサーバから取得すればよいいため、呼び出しが成功するまで各 JNDI サーバに対して順次呼び出す。

#### 名前の参照



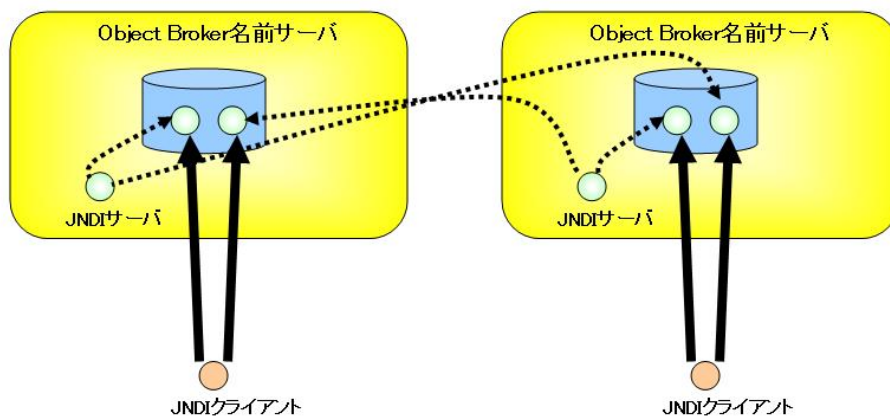
この動作によって各 JNDI サーバに登録されている情報が同じになります。

これらの処理は WebOTX の提供する JNDI SPI(Service Provider Interface)クラスの中で行われますので、アプリケーションでは意識する必要はありません。

HTTP セッションレプリケーションでは Web コンテナが JNDI サーバに対して HTTP セッションオブジェクトを登録、参照します。

## 4 JNDI サーバのレプリケーションの設定方法

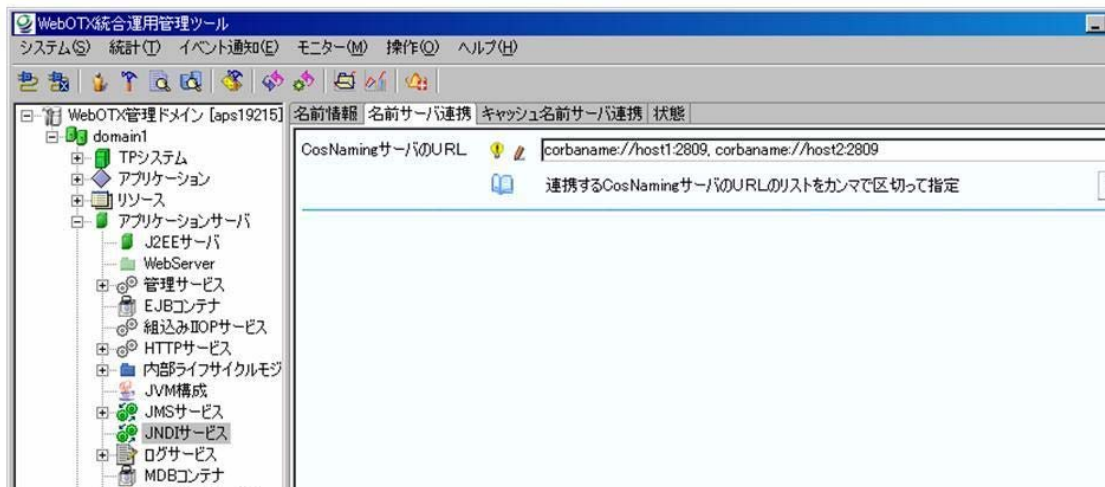
JNDI サーバをレプリケーションさせるための具体的な設定方法について説明します。基本的には各ドメインの Object Broker 名前サーバには同期される全ての JNDI サーバのオブジェクトが登録するように設定します。これによってどのドメインにアクセスしても他の JNDI サーバと同期されます。



JNDI サーバのオブジェクトの登録先の指定方法は以下のとおりです。

### 1. 運用管理ツールを使用する場合

アプリケーションサーバ→ JNDI サービスの「名前サーバ連携」に登録する Object Broker 名前サーバの `corbaname` 形式 URL を指定します。登録先コンテキスト名(“`NEC/JNDI`”)は自動的に付加されるので URL に含める必要はありません。複数の名前サーバに登録するときは URL をカンマで区切って記述します



## 2. 運用管理コマンドを使用する場合

set コマンドで運用管理ツールに指定する URL を同様に指定します。

**otxadmin> set server.jndi-service.url=<登録する名前サーバの URL>**

基本的にはこれだけ設定して WebOTX のドメインまたは JNDI サービスのみを再起動すれば以後の JNDI サーバに対する登録、削除は全ドメインの JNDI サーバに対して行われます。

## 5 JNDI サーバとの通信障害の対処方法について

JNDI サーバの障害対応として、最も事例の多い通信障害の対処方法について説明します。JNDI の呼び出しが以下の例外により失敗する場合、Object Broker 名前サーバから JNDI サーバのオブジェクトが取得することができないことが原因です。

**javax.naming.CommunicationException: Can't find JNDI name server. ...**

以下のこのエラーが発生したときの原因と対処について説明します。

### 5.1. Object Broker サービスとの通信に失敗している場合

“Can't find JNDI name server. ...” のメッセージの後に org.omg.CORBA パッケージの

Java 例外が出力されている場合は、Object Broker 名前サーバと通信する際にエラーが発生しています。

以下の場合には Object Broker の名前サーバが起動していないことが原因です。

**org.omg.CORBA.NO\_RESPONSE: Initial reference for NameService not found. (<ホスト名>) jp.co.nec.orb.Ob.OadPackage.NotFound: IDL:Ob/Oad/NotFound:1.0 vmcid: 0x1000 minor code: 996 completed: No ...**

また、Object Broker の名前サーバと通信するときは Object Broker の oad というプロセスとも通信を行います。その oad が動作していないときは以下のようなエラーが発生します。

**org.omg.CORBA.COMM\_FAILURE: java.net.ConnectException: Connection refused: connect:<ホスト名>: <oad ポート番号> vmcid: 0x1000 minor code: 911 completed: No ...**

このような場合は、動作していない Object Broker のサービスの動作に問題が無いかを確認してください。よくあるケースとしては Object Broker サービスが使用するポートが他のプロセスで使用されているために起動に失敗しています。

## 5.2. JNDI サーバとの通信に失敗している場合

org.omg.CORBA パッケージのエラーではなく、“**please check name server is active**”というメッセージが出力されたときは、Object Broker 名前サーバとの通信は成功したが、そこから接続可能な JNDI サーバのオブジェクトが一つも取得できなかったことが原因です。以下の点を確認してください。

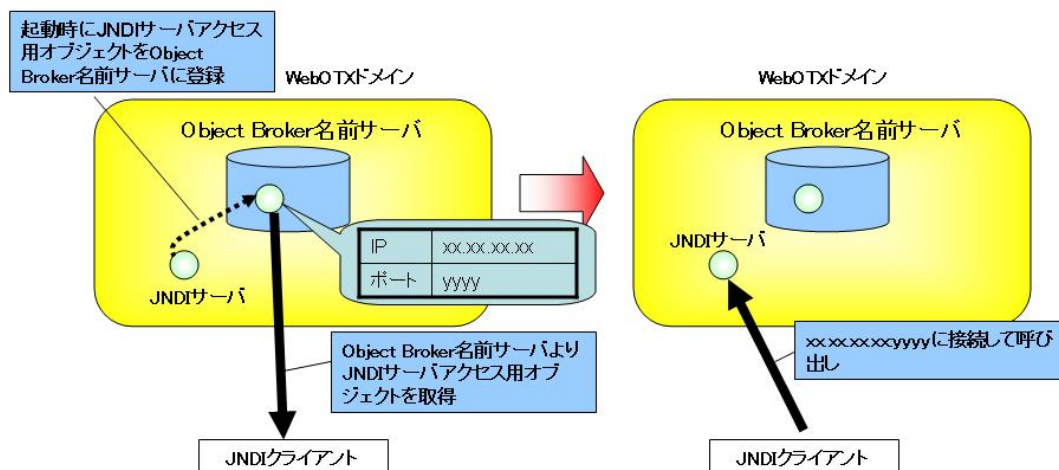
1. 対象となる WebOTX ドメインの JNDI サービスが起動しているか
2. JNDI クライアント側の `java.naming.provider.url` プロパティが正しく設定されているか。
- 3 JNDI サーバの通信先アドレスがクライアントから見えないアドレスとなっていないか

以下 3 番目のケースについて詳しく説明します。

JNDI サーバに接続するときの IP アドレス、ポートの情報は Object Broker 名前サーバに登録されるリモートオブジェクトの中に格納され、クライアントはその情報を元に接続を試みます。IP アドレスには WebOTX のデフォルト設定ではサーバ側で Java のローカルホ



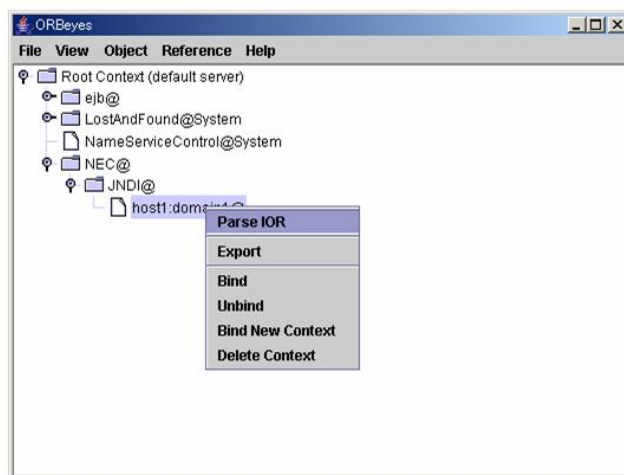
ストの IP アドレス取得 API を呼び出した結果が入ります。サーバが複数のネットワークインタフェースを持つ場合、どれが使用されるかは OS の設定に依存しますので、クライアントから見えないアドレスが設定されてしまう可能性があります。



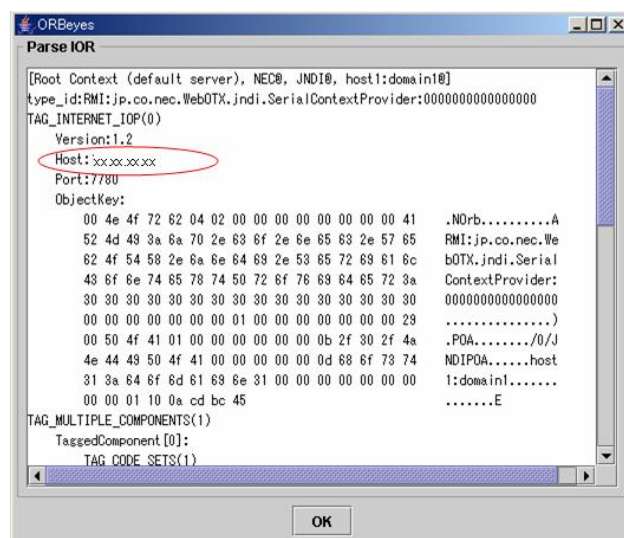
これが原因となっているかを調査するためには JNDI サーバのリモートオブジェクトに埋め込まれる IP、ポートの情報を確認する必要がありますが、それには WebOTX の Web サイトで公開されている「ORBeyes」または「BindingViewer」というツールを使用します。(BindingViewer は ORBeyes のコマンドライン版です)。

## 1. ORBeyes を使用する場合

対象となる WebOTX の Object Broker 名前サーバに接続して “NEC/JNDI” 下に登録されている名前を選択し、右クリックメニューから「ParseIOR」を選択します。



その名前のオブジェクトの内容が別ウィンドウで表示されます。  
「TAG\_INTERNET\_IOP(0)」の「Host」が接続先の IP アドレスとなりますので、これがクライアントからアクセス可能であるか確認してください。



## 2. BindingViewer を使用する場合

以下のように実行します。ポートは対象となる WebOTX ドメインで動作している Object Broker 名前サーバのポート番号を指定します。(インストール時に作成される domain1 の場合、ポート番号は 2809 となります)

**BindingViewer -namesv:<ホスト>:<ポート> -select:h**

結果はコンソールに出力されます。その中から以下のように NEC/JNDI の下に登録されている情報に着目し、Host の値がクライアントからアクセス可能な IP アドレスとなっているか確認してください。

L ctx, ID:NEC, Kind:

L ctx, ID:JNDI, Kind:

L obj, ID:host1:domain1, Kind:, Host:xx.xx.xx.xx, Port:7780

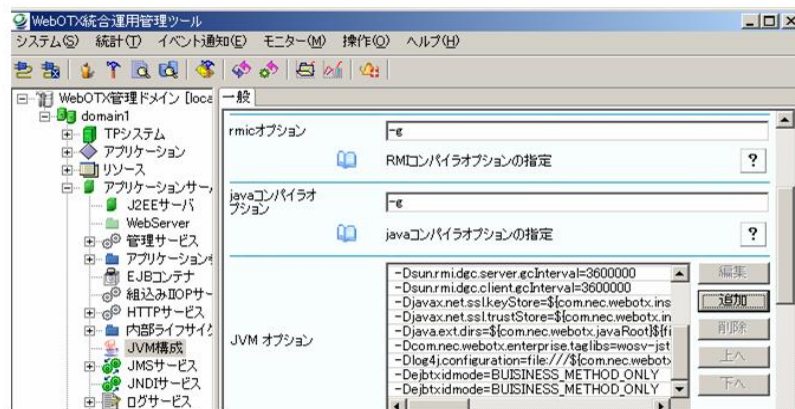
確認した結果、JNDI サーバの IP アドレスがクライアントからアクセス可能なアドレスとなっていなかったときは、WebOTX JMX エージェントプロセスに対して以下の java オプションを指定してオブジェクトに埋め込まれる IP アドレスを変更します。(設定後ドメイン再起動が必要です)

**-Djp.co.nec.orb.ServerAddr=<IP アドレス>**

以下具体的な設定方法について説明します。

## 1. 運用管理ツールを使用する場合

「アプリケーションサーバ」→「JVM 構成」の「JVM オプション」の「追加」ボタンを押して上記オプションの指定を追加します。



## 2. 運用管理コマンドを使用する場合

対象の WebOTX ドメインと接続して以下のコマンドを実行します。

**otxadmin> create-jvm-options -Djp.co.nec.orb.ServerAddr=<IP アドレス>**