



# User Guide

## OEC API Remoting Setup

April 17, 2008



---

<b>INTRODUCTION .....</b>	<b>3</b>
<b>DOCUMENT HISTORY .....</b>	<b>3</b>
<b>DESCRIPTION .....</b>	<b>4</b>
<b>PROCEDURES .....</b>	<b>6</b>
<b>USAGE .....</b>	<b>6</b>
<b>GLOSSARY .....</b>	<b>7</b>

## Introduction

The purpose of this document is to describe OEC API Remote Connection – the capability to share a single API instance among several software applications.

The OEC System does not support multiple connections from the single user (login name). However, some clients use distributed frameworks for separate order and price processing. The OEC API Remote Connection avoids using a secondary login name and downloading the same data twice. Also, it does not require significant changes in the client code – so multiple client applications share the OEC API instance exactly as before.

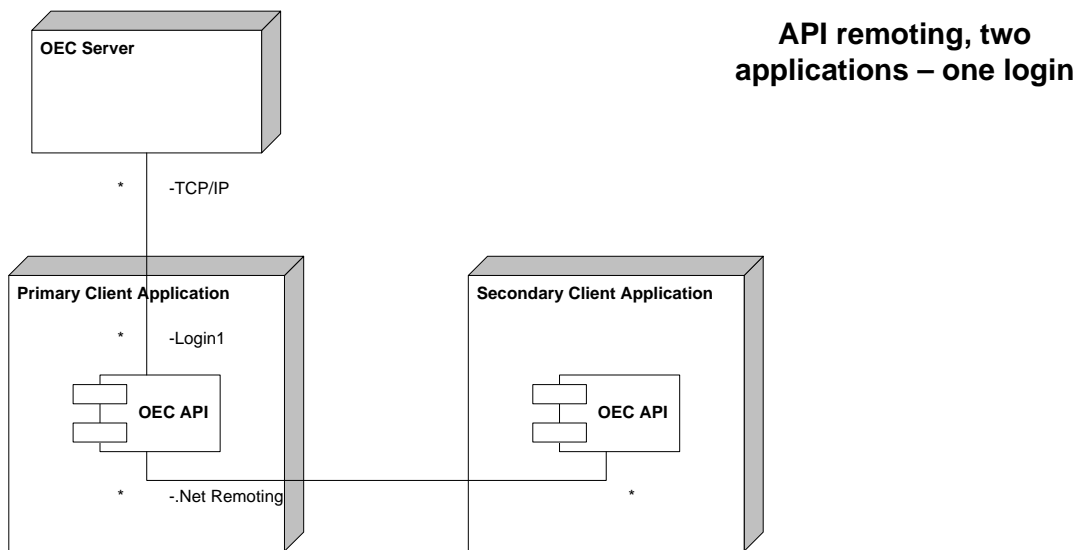
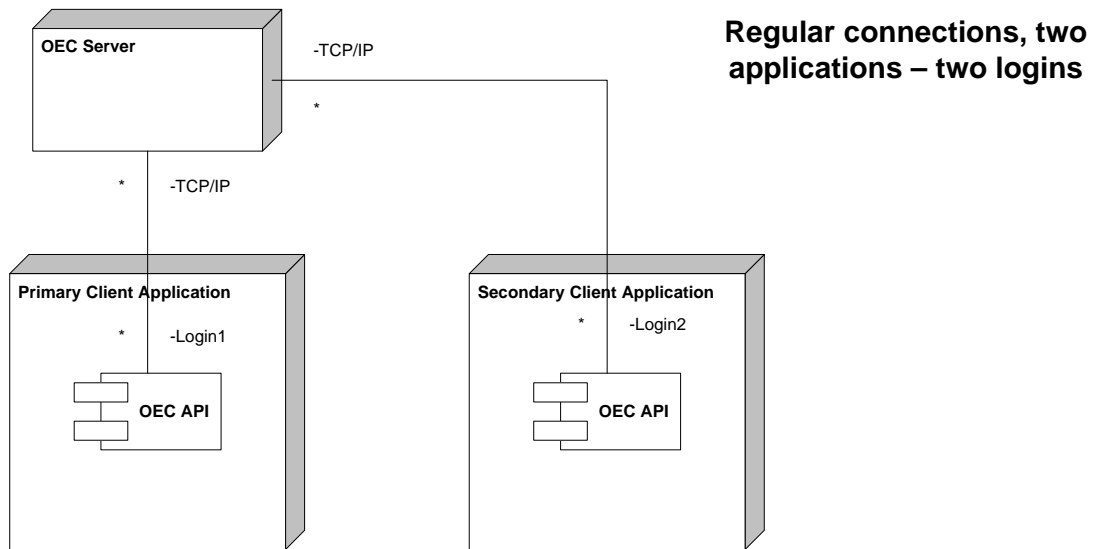
## Document History

Version	Effective Date	Authorization
OEC Trader 3.1.2	4/16/08	Information Technology

*Note: This document supports all previous versions through the effective date of this release.*

## Description

The OEC API Remote is implemented using .NET Remote for sharing the OEC Client instance between client processes. When the first application creates an API instance, it serves as a .NET Remote server for subsequent clients and allows them to connect to the existing instance and share functionality.



## Procedures

### Usage

There are multiple options to implement the OEC API in three modes:

1. This is the regular instance that is used by hosted application only. (Remote disabled)

```
OEC.API.OECClient OECClient = new OEC.API.OECClient();
```

2. This is the regular instance that is used by the hosted application directly and exposes the remote server for others (“primary client application”).

```
OEC.API.OECClient OECClient = new OEC.API.OECClient();
```

```
OECClient.RemoteHostingEnabled = true;
```

3. This is the OEC Remote instance that connects to the exposed API of the primary application.(Remote enabled)

```
OEC.API.OECClient OECClient = OEC.API.OECClient.CreateInstance(true);
```

Post-condition: `OECClient.RemoteHostingEnabled` is `true`, if it is remoting instance, `false` – if there is no exposed OEC API instance on the box and the regular instance has been created.

Note: a call `OEC.API.OECClient.CreateInstance(false)` is identical to `new OEC.API.OECClient();`

## Glossary

All terminology and related operational concepts for these processes are listed in ascending order. The acronyms are within parentheses after the entry.

Term/Acronym	Definition
Application Program Interface (API)	This refers to a set of routines, protocols, and tools for building software applications.*
remote	In networks, <i>remote</i> refers to files, devices, and other resources that are not connected directly to the workstation. *

\*www.Webopedia.com