

# Allgemeine Informationen

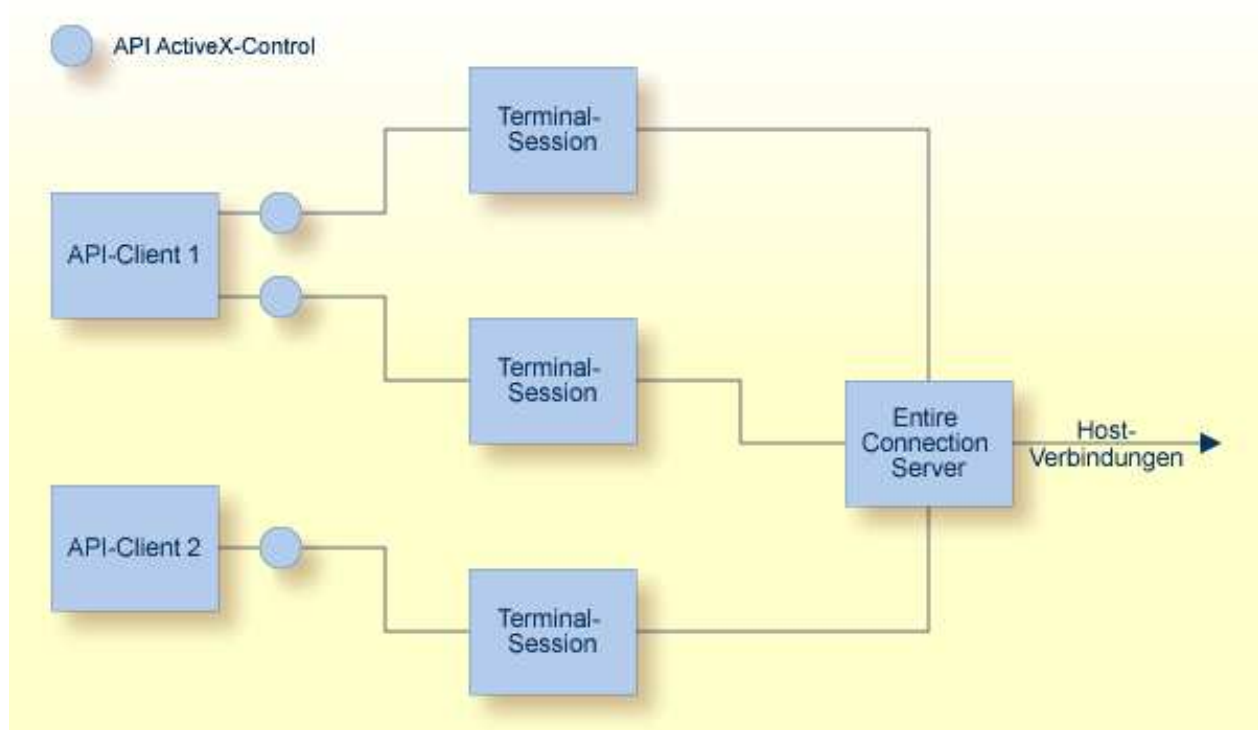
Dieses Kapitel behandelt die folgenden Themen:

- API-Controls und Terminal-Sessions
- Synchrone und asynchrone Aufrufe
- Glossar

---

## API-Controls und Terminal-Sessions

Jedes ActiveX kann ein einzelnes Terminal von Entire Connection unterstützen. Deshalb kann eine einzelne Anwendung mit beliebig vielen Terminal-Sessions arbeiten, wobei viele Controls gleichzeitig aktiv sind.



Jedes API-Control kann die Verbindung zu einer bestehenden Terminal-Session herstellen oder eine neue Terminal-Session erstellen. Jede Terminal-Session kann jederzeit mit einem API-Control verbunden sein. Die einzige Ausnahme ist der UA-Modus, bei dem das Verbinden nicht erlaubt ist. Außerdem ist es nicht möglich, den UA-Modus bei einem Terminal einzuschalten, das durch die Programmierschnittstelle kontrolliert wird.

Im API-Modus ist eine Terminal-Session in der Regel unsichtbar, um Benutzereingaben zu verhindern. Wenn das Terminal durch die Programmierschnittstelle sichtbar gemacht wird, hat der Benutzer die volle Kontrolle über das Terminal. Unter anderem kann er dann Prozedurdateien ausführen und die Terminal-Session beenden. Jeder Datentransfer und jede Prozedurdatei bleibt unter der Kontrolle des API-Clients.

## Synchrone und asynchrone Aufrufe

Synchrone (blockierende) und asynchrone (nicht-blockierende) Aufrufe stehen mit Visual Basic .NET, C++ und C# zur Verfügung. Während der Planungsphase entscheiden Sie, welche dieser beiden Aufrufarten für Ihre Zwecke am Besten geeignet ist.

Im asynchronen Modus kehren fast alle API-Aufrufe sofort mit einem entsprechenden Return-Code zurück. Ausnahmen sind die Initialisierungsfunktionen und die Funktionen zum Beenden der Verbindung mit einer Terminal-Session. Diese Funktionen blockieren immer, unabhängig vom gewählten Modus.

Nachdem ein Befehl bei der asynchronen API-Ausführung abgearbeitet wurde, sendet das Control ein Ereignis, das bestätigt, dass der Befehl abgearbeitet wurde. Die Parameter dieses Ereignisses enthalten das Ergebnis des Aufrufs (d.h. eine Rückmeldung wie der Befehl abgearbeitet wurde) und alle angeforderten Daten.

Die Beschreibungen in der *Übersicht der API-Aufrufe* informieren Sie darüber, ob ein Aufruf nur synchron ausgeführt werden kann. In allen anderen Fällen wird ein Ereignis gesendet, das bestätigt, dass der Befehl abgearbeitet wurde; `LogonEntireConnection` sendet zum Beispiel `LogonComplete`.

In bestimmten Situationen erzeugt das API-Control auch Benachrichtigungsereignisse ohne Berücksichtigung des Modus, in dem es gerade ausgeführt wird. Dies können Fehlermeldungen, Informationen und Datentransferdaten sein.

## Glossar

|                  |   |
|------------------|---|
| API              | Funktionalität, die Drittanbieteranwendungen zur Verfügung steht.   |
| API-Client       | Die Anwendung, die Entire Connection mit Hilfe der Programmierschnittstelle kontrolliert.   |
| API-Control      | Das vom API-Client verwendete ActiveX.  |
| Terminal-Session | Die Terminal-Anwendung von Entire Connection.   |
| Asynchron        | Nicht-blockierender Modus. Die Programmierschnittstelle kehrt sofort zu der aufrufenden Anwendung zurück. Wenn die Verarbeitung abgeschlossen ist, sendet die Programmierschnittstelle eine Nachricht an die Anwendung. |
| Synchron         | Die Programmierschnittstelle kehrt nur dann zu der aufrufenden Anwendung zurück, wenn die Verarbeitung des Funktionsaufrufs abgeschlossen ist.  |