



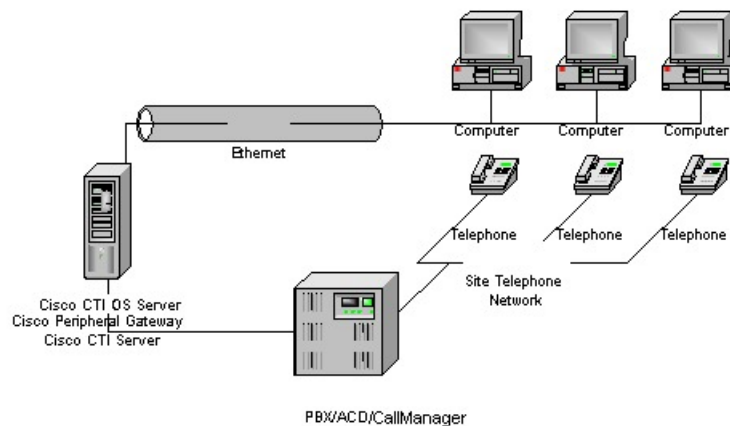
## Introduction

- [Overview of CTI OS, on page 1](#)
- [System Manager Responsibilities, on page 3](#)
- [System Requirements, on page 3](#)

## Overview of CTI OS

CTI OS combines a powerful, feature-rich server and an object-oriented software development toolkit to enable rapid development and deployment of complex CTI applications. Together, the Cisco CTI Server Interface, CTI OS Server, and CTI OS Client Interface Library (CIL) create a high performance, scalable, fault-tolerant three-tiered CTI architecture, as illustrated in following figure.

**Figure 1: CTI OS Three-Tiered Architecture Topology**



The CTI OS application architecture employs three tiers:

- The CIL is the first tier, providing an application-level interface to developers.
- The CTI OS Server is the second tier, providing the bulk of the event and request processing and enabling the object services of the CTI OS system.

- The Cisco CTI Server is the third tier, providing the event source and the back-end handling of telephony requests.

## Advantages of CTI OS as Interface to Unified ICM Enterprise

CTI OS brings several major advances to developing custom CTI integration solutions. The CIL provides an object-oriented and event-driven Application Programming Interface (API), while the CTI OS Server does the *heavy-lifting* of the CTI integration: updating call context information, determining which buttons to enable on softphones, providing easy access to supervisor features, and automatically recovering from failover scenarios.

The key advantages of CTI OS include:

- **Rapid integration.** Developing CTI applications with CTI OS is easier and faster than any previously available Cisco CTI integration platform. The same object-oriented interface is used across programming languages, enabling rapid integrations in C++, Visual Basic, .NET, Java, or any Microsoft COM-compliant container environment.



---

**Note** The inclusion of the .NET toolkit allows for custom applications written in C#, VB.NET, or any other CLR-compliant language. By starting with the code for the .NET sample, the CTI Toolkit Combo Desktop developers can quickly customize the code without having to start from scratch.

---

CTI OS enables developers to create a screen-pop application in as little as five minutes. The only custom-development effort required is within the homegrown application to which you add CTI.

- **Complex solutions made simple.** CTI OS enables complex server-to-server integrations and multiple agent monitoring-type applications. The CIL provides a single object-oriented interface that you can use in two modes: agent mode and monitor mode. For more information about these two modes, see CTI OS Developer Guide for Cisco Unified ICM at: [https://www.cisco.com/en/US/products/sw/custcosw/ps14/products\\_programming\\_reference\\_guides\\_list.html](https://www.cisco.com/en/US/products/sw/custcosw/ps14/products_programming_reference_guides_list.html).
- **Fault tolerant.** CTI OS is built upon the Unified ICM Node Manager fault-tolerance platform, which automatically detects process failure and restarts the process, enabling work to continue. Upon recovery from a failure, CTI OS initiates a complete, system-wide snapshot of all agents, calls, and supervisors and propagates updates to all client-side objects.

## Key Benefits of CTI OS for CTI Application Developers

The CTI OS CIL provides programmers with the tools required to rapidly develop high-quality CTI-enabled applications, taking advantage of the rich features of the CTI OS Server. Every feature of CTI OS was designed with ease of integration in mind, to remove the traditional barriers to entry for CTI integrations:

- **Object-oriented interactions.** CTI OS provides an object-oriented CTI interface by defining objects for all call center interactions. Programmers interact directly with Session, Agent, SkillGroup, and Call objects to perform all functions. CIL objects are thin proxies for the server-side objects, where all the 'heavy-lifting' is done. The Session object manages all objects within the CIL. A UniqueObjectID identifies each object. Programmers can access an object by its UniqueObjectID or by iterating through the object collections.

- **Connection and session management.** The CTI OS CIL provides out-of-the-box connection and session management with the CTI OS Server, hiding all of the details of the TCP/IP sockets connection. The CIL also provides out-of-the-box failover recovery. Upon recovery from a failure, the CIL automatically reconnects to another CTI OS Server (or reconnects to the same CTI OS Server after restart), reestablishes the session, and recovers all objects for that session.
- **All parameters are key-value pairs.** The CTI OS CIL provides helper classes to treat all event and request parameters as simply a set of key-value pairs. All properties on the CTI OS objects are accessible by name via a simple `Value = GetValue("key")` mechanism. Client programmers can add values of any type to the CTI OS Arguments structure using the enumerated CTI OS keywords or their own string keywords (for example, `AddItem["DialedNumber", "1234"]`). This provides for future enhancement of the interface without requiring any changes to the method signatures.
- **Simple event subscription model.** The CTI OS CIL implements a publisher-subscriber design pattern to enable easy subscription to event interfaces. Programmers can subscribe to the event interface that suits their needs, or use the AllInOne interface to subscribe to all events. Subclassable event adapter classes enable programmers to subscribe to event interfaces and only add minimal custom code for the events they use, and no code at all for events they do not use.

## System Manager Responsibilities

The remainder of this document provides step-by-step procedures for the tasks a system manager must perform to set up and configure CTI OS. These tasks include:

- Installing CTI OS Server.
- Installing CTI Toolkit Agent Desktop, Supervisor Desktop, Tools, Documentation, Win32 SDK, Java SDK, and .NET SDK.
- Enabling CTI OS security.
- Using the Windows Registry Editor (`regedit.exe`) to configure the required CTI OS registry keys.
- Starting CTI OS and its associated processes from Unified CCE Service Control.



---

**Note** You *must* have administrator privileges to perform the procedures discussed in this manual.

---

## System Requirements

See the *Solution Design Guide for Cisco Unified Contact Center Enterprise* at [http://www.cisco.com/en/US/products/sw/custcosw/ps1844/products\\_implementation\\_design\\_guides\\_list.html](http://www.cisco.com/en/US/products/sw/custcosw/ps1844/products_implementation_design_guides_list.html)

For more information on system requirements, see the *Virtualization for Unified Contact Center Enterprise* at [http://www.cisco.com/c/dam/en/us/td/docs/voice\\_ip\\_comm/uc\\_system/virtualization/virtualization-unified-contact-center-enterprise.html](http://www.cisco.com/c/dam/en/us/td/docs/voice_ip_comm/uc_system/virtualization/virtualization-unified-contact-center-enterprise.html)

## Set User Privileges

On supported Windows client machines, users must have privileges that enables them to run legacy applications and have read/write access to the Cisco registry keys that the desktop applications use. To set user privileges to enable users to run CTI OS Agent Desktop and CTI OS Supervisor Desktop, an administrator must perform the following steps.

### Procedure

---

- Step 1** On the Microsoft Windows Start Menu, select **Start > Run**.
- Step 2** Type in **regedt32** and click **OK**.  
The Microsoft Windows Registry Editor window appears.
- Step 3** Go to the following registry location:  
`HKEY_LOCAL_MACHINE\SOFTWARE\Cisco Systems, Inc.\CTI Desktop\Ctios`
- Step 4** Select **Security > Permissions**.  
A Permissions dialog box appears.
- Step 5** If you are adding a new user, perform the following steps.
- Click **Add**.  
A Select Users dialog box appears.
  - Select the user to be added from the list in the top half of the Select Users dialog box.
  - Click **Add**, then click **OK**.  
You return to the Permissions dialog box; the user you just added is now on the list.
- Step 6** Click the user whose privileges you want to set.
- Step 7** Set the Full Control permissions for this user to **Allow**.
- Step 8** Click **Apply**.
- Step 9** Click **OK**.
- Step 10** Exit Registry Editor.
-