

Agent Login Flow:

The Agent Login and CTI monitoring in UCCE solution is controlled by below Agent PG components:

**CTI Server:**

CTI server provide CTI interface to UCCE solution through Agent PG for Agent State request like Login/Logout/Make Call/Answer Call/ Transfer/Conference/Hold etc.

Some Clients Connects to CTI server Directly like Cisco Finesse, third party CRM connectors etc.

Some clients uses CTI OS server interface to connect to CTI server. Those clients are CTI-OS Toolkit desktop, CAD etc.

CTIOS is Cisco's next generation customer contact integration platform. CTIOS combines a powerful, feature-rich server and an object-oriented software development toolkit to enable rapid development and deployment of complex CTI applications in Java, .NET, COM or C++. Together with the Cisco CTI Server Interface, CTIOS and Client Interface Library (CIL) create a high performance, scalable, fault-tolerant three-tiered CTI architecture.

**PIM (Peripheral Interface monitor) Process:**

Agent PG's PIM process is a process which connects to CUCM CTI manager service for Bridging communication between UCCE Central controller and CUCM server.

It converts messages from both side which can be understood by each sides.

PIM uses Application User created on CUCM to login to CUCM with standard CTI roles.

And also it can monitor and control device which are added to application users controlled device list.

**JGW (JTAPI Gateway):**

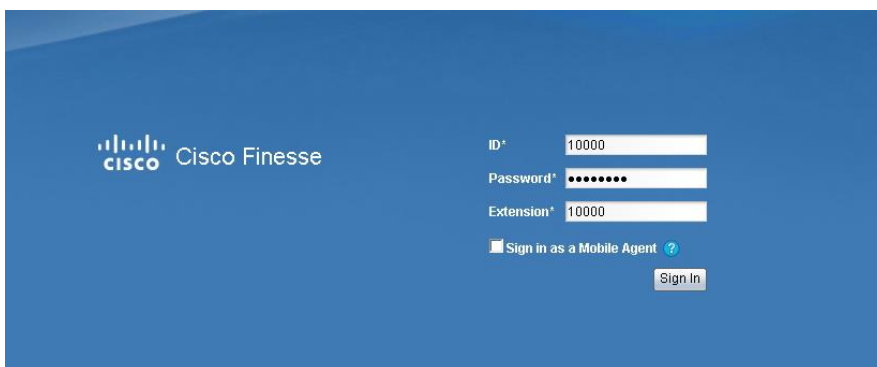
JGW process is a JTAPI gateway process runs on Agent PG and in conjunction with Agent PIM, both process are useless without each other.

JGW gateway forms actual interface with CUCM and PIM.

Note: there are other process also exist like OPC and MDS on Agent PG, which enables interposes communications. E.g. PIM process can communicate Central Controller and CTI Server and Vice Versa.

Now let's look at what event flows when agent tries to login to Finesse:

1. Agent Logs into Finesse with Extension and proper credentials:



2. Now with Finesse architecture, the login information (Agent ID/Password) will be authenticated with AW, finesse connects to AW database over standard TCP/IP connection.
3. With Successful Authentication, Finesse Submits login Request to CTI server, it's connected to:

```
23:35:36:354 cg1A-ctisvr Trace: *** AddToAssociateAgentList();      ADDED: SessionID=1 AgentID=10000 PeripheralID=5000
23:35:36:354 cg1A-ctisvr Trace: CSTASetAgentState: InvokeID=0x2600001d Dev=10000 AgentMode=LOG_IN AGID=10000 SG=-
1(0xffffffff)
23:35:36:354 cg1A-ctisvr Trace: PrivateData: EventReasonCode=50004 WorkMode=0 NumAdditionalGroups=0 PositionID=
SupervisorID= ClientAddress=
23:35:36:470 cg1A-ctisvr Trace: Server::ActivateAgentConfigureUpdateTimer
23:35:36:470 cg1A-ctisvr Updated 1 Agents.
23:35:36:521 cg1A-ctisvr Trace: AGENT_EVENT: ID=10000 Periph=5000 Ext=10000 Inst=10000 Sig=
23:35:36:521 cg1A-ctisvr Trace: SkgState=LOGIN SkgDuration=0 OverallState=NOT_READY OverallDuration=0 Reason=0
23:35:36:521 cg1A-ctisvr Trace: MRDID=1 NumTasks=0 MaxTaskLimit=1 AgtMode=1 AGTAvailabilityStatus=0 ICMAgtID=5001
23:35:36:521 cg1A-ctisvr Trace: SkTgtID=5001 SkGrpNo=0x9324 SkGrpID=5000 NumLines=0 CurLine=0 ClientStatus=0x0 Direction=0
23:35:36:521 cg1A-ctisvr Trace: SetAgentDevice: Agent(10000) -> Device(10000).
```

4. CTI Server will pass the request to OPC:

```
23:35:36:354 PG1A-opc Trace: TPServices::AddInvocationByCTIInvokeID - CTIInvokeID 0x2600001d mapped to OPCInvokeID 0xf000001b
23:35:36:354 PG1A-opc Trace: CSTASetAgentState: TPServices::AddInvocationByCTIInvokeID InvokeID=0xf000001b Dev=10000 AgentMode=LOG_IN AGID=10000 SG=-1(0xffffffff)
23:35:36:354 PG1A-opc Trace: PrivateData: EventReasonCode=50004 WorkMode=0 NumAdditionalGroups=0 PositionID= SupervisorID= ClientAddress=
```

5. OPC will Request Pim to process the login request, here PIM will check if device target exist for particular extension and if yes then only the login action will be allowed:

```
23:35:36:355 PG1A-pim1 Trace: CSTASetAgentState: Peripheral::ProcessCSTAThirdPartyServiceRequest InvokeID=0xf000001b Dev=10000 AgentMode=LOG_IN AGID=10000 SG=-1(0xffffffff)
23:35:36:355 PG1A-pim1 Trace: PrivateData: EventReasonCode=50004 WorkMode=0 NumAdditionalGroups=0 PositionID= SupervisorID= ClientAddress=
23:35:36:355 PG1A-pim1 Trace: CSTASetAgentState: localDeviceId: 10000 -- remoteDeviceID:
23:35:36:355 PG1A-pim1 Trace: PimConfig::IsExtConfiguredInATR - Found the extension 10000 in a valid ATR range
23:35:36:355 PG1A-pim1 Trace: Generated network target id is:100003
23:35:36:355 PG1A-pim1 Trace: DeskLinkDeviceTarget::Constructor: NetworkTargetID=100003, Type=CISCOPHONE, Ext=10000, Dialed#=10000
23:35:36:355 PG1A-pim1 Trace: TelephonyDriver:Login IP address Login Device ID 10000 Device Target Cisco10000 Instrument 10000
23:35:36:355 PG1A-pim1 Trace: ActivateJTAPIClient: sending msgAddCallObserver
```

6. Pim will request JGW process to add call observer for particular extension, the call observer is nothing but monitoring and controlling session JGW can established for extension. JGW process also gets to know here about the device where line information is associated with, IP addressing mode, device busy and retry configuration etc. The login can be rejected if Device or line is not configured properly here.

```
23:35:36:355 PG1A-jgw1 Trace: MsgAddCallObserver: Addr: 10000 Remote Addr: 0 InvID: 277865 CallDeliveryMode ID: 0.
23:35:36:355 PG1A-jgw1 Trace: Adding Call Observer to: 10000.
23:35:36:355 PG1A-jgw1 Trace: Address Name: 10000IP Addressing Mode:IP_ADDRESSING_MODE_IPV4_V6.
23:35:36:355 PG1A-jgw1 Trace: handleMsgAddCallObserver::Before addCallObserver for Addr: 10000 RegState: InService.
23:35:36:357 PG1A-jgw1 Trace: Adding Terminal Observer to: 10000.
23:35:36:357 PG1A-jgw1 Trace: Call to addCallObserver and address.getTerminals for 10000 returned in 2 milsecs.
23:35:36:357 PG1A-jgw1 Trace: CiscoTermInServiceEv Term: CIPCWIN7B Cause: NORMAL.
23:35:36:358 PG1A-jgw1 Trace: Call to CiscoAddress[10000].getAddressCallInfo() returned in 1 milsecs.
23:35:36:358 PG1A-jgw1 Trace: MsgAddCallObserverResponse: Addr: 10000 Succeeded: 1 InvID: 277865 Cause: -1.
```

7. After establishing successful call observer PIM is notified about it:

```
23:35:36:358 PG1A-pim1 Trace: ApplicationProtocol::RecvAddCallObserverResponseMsg: Address: 10000 Succeeded: 1
Invid=277865 cause=-1
23:35:36:358 PG1A-pim1 Trace: AgentAttributeLoginReqMSG invokeID 0xf000001b
23:35:36:358 PG1A-pim1 Trace: RecvSnapshotDeviceResponseMsg: SnapshotDeviceResponse (92); Message Length 55 bytes
Invoke ID: (0) 00000000
Succeeded: (0) 00000000
Calling Device ID: 10000
ErrorCode: (0) 00000000
CallCount: (1) 00000001
CID: Undefined
ConnDevType: (0) 00000000
ConnDev:
LineType: (0) 00000000
LocalConnInfo: (0) 00000000
SilentMonitorStatus: (0) 00000000
COCCallID: Undefined
COCCDeviceID:
COCCDeviceIDType: (0) 00000000

23:35:36:469 PG1A-pim1 ADDED\UPDATED 1 Agents on peripheral: 5000.
```

8. Pim Notifies OPC, OPC processes the request and informs Central controller about the login also passes the dynamic object information to central controller about agent Id and Extension association so central controller can use it for routing. Agent login request will be processed for each skill group and PQ.

```
23:35:36:359 PG1A-opc Trace: Peripheral::ProcessVoiceAgentAttributeLoginReq - AgentSTID=5001 on Peripheral=5000
```

From Router logs:

```
23:35:36:404 ra-rtr Trace: AgentAttrLoginReq: Agent=5001 (agent1), Peripheral=5000 (CUCM_PIM1)
23:35:36:404 ra-rtr Trace: Agent 5001(agent1) - PQSteps,5000(Englsh_Spanish.1),5001(Englsh_Spanish.2),
23:35:36:404 ra-rtr Trace: AgentSGAssignment : Agent 5001(agent1) PQChange 1, SG's - 5006
23:35:37:204 ra-rtr Trace: Agent 5001 (10000) on peripheral 5000 (CUCM_PIM1) logged in to group 5000.
23:35:37:204 ra-rtr Trace: Agent 5001 (10000) on peripheral 5000 (CUCM_PIM1) logged in to group 5002.
23:35:37:204 ra-rtr Trace: Agent 5001 (10000) on peripheral 5000 (CUCM_PIM1) logged in to group 5003.
23:35:37:204 ra-rtr Trace: Agent 5001 (10000) on peripheral 5000 (CUCM_PIM1) logged in to PQ 5000.
```

9. CTI server and Finesse are notified and agent login is successful:

```
23:35:36:523 cg1A-ctisvr Trace: CSTASetAgentStateConfEvent: InvokeID=0x2600001d
23:35:36:523 cg1A-ctisvr Trace: AGENT_EVENT: ID=10000 Periph=5000 Ext=10000 Inst=10000 Sig=Finesse
23:35:36:523 cg1A-ctisvr Trace: SkgState=UNKNOWN SkgDuration=0 OverallState=NOT_READY OverallDuration=0 Reason=0
23:35:36:523 cg1A-ctisvr Trace: MRDID=1 NumTasks=0 MaxTaskLimit=1 AgtMode=1 AGTAvailabilityStatus=0 ICMAgtID=5001
23:35:36:523 cg1A-ctisvr Trace: SkTgtID=5001 SkGrpNo=0xffffffff SkGrpID=-1 NumLines=0 CurLine=0 ClientStatus=0x1 Direction=0
23:35:36:523 cg1A-ctisvr Trace:
23:35:36:523 cg1A-ctisvr Trace: SET_CLIENT_STATUS_RESP: InvokeID=0x2600001e ResultCode=0(OK)
```

So At High level, The initial login will be processed by Agent PG and CTI server and if login is successful the Dynamic object is created for Agent ID and Extension and passed back to Central controller, central controller can now use this information for call routing.

Agent Targeting Rules are required to enable Agent Login and Call Routing for Agent labels, it's just configuration piece but matters a lot.