


# Glossary - ADK Terms and Concepts

---

**Description:** Comprehensive glossary of Google Agent Development Kit (ADK) terms, concepts, and terminology used throughout the tutorials.

# Glossary - ADK Terms and Concepts

---

 **Purpose:** Comprehensive reference for Google Agent Development Kit (ADK) terminology and concepts used throughout the tutorials.

 **Source of Truth:** [google/adk-python](https://github.com/google/adk-python) (<https://github.com/google/adk-python>) (ADK 1.15) + Official Google Documentation

---

## A

---

### | Agent

A complete AI system powered by a Large Language Model (LLM) that can perform tasks through tools, maintain state, and interact with users. Agents are more than just LLMs - they include reasoning, tools, memory, and instructions.

**See Also:** [Tutorial 01: Hello World Agent](#) ([01\\_hello\\_world\\_agent.md](#))

---

## | Agent-to-Agent (A2A) Communication

Protocol for agents to communicate and collaborate with each other, enabling distributed multi-agent systems.

**See Also:** [Tutorial 17: Agent-to-Agent Communication](#) (17\_agent\_to\_agent.md)

## | Agent Engine

Google Cloud's managed service for deploying and scaling agents on Vertex AI, providing built-in scaling, monitoring, and version management.

**See Also:** [Tutorial 23: Production Deployment](#) (23\_production\_deployment.md)

## B

---

## | Built-in Tools

Pre-built tools provided by Google ADK for common operations like web search, location services, and code execution.

**See Also:** [Tutorial 11: Built-in Tools & Grounding](#) (11\_built\_in\_tools\_grounding.md)

## C

---

## | Callbacks

Functions that execute at specific points in an agent's lifecycle (before/after agent runs, tool calls, etc.) for monitoring, guardrails, and control flow.

**See Also:** [Tutorial 09: Callbacks & Guardrails](#) (09\_callbacks\_guardrails.md)

## | Context Window

The maximum amount of text (measured in tokens) that an LLM can process at once. Exceeding this limit causes errors.

## | CopilotKit

React component library for building AI chat interfaces that integrate with ADK agents.

## E

---

## | Evaluation

Systematic testing and quality assessment of agent behavior using automated metrics and human review.

**See Also:** [Tutorial 10: Evaluation & Testing](#) (10\_evaluation\_testing.md)

## | Events

Structured logging system that tracks agent execution, state changes, tool calls, and errors for debugging and monitoring.

**See Also:** [Tutorial 18: Events & Observability](#) (18\_events\_observability.md)

## F

---

## | Function Tools

Regular Python functions that agents can call to perform specific tasks. ADK automatically generates schemas from function signatures and docstrings.

**See Also:** [Tutorial 02: Function Tools](#) (02\_function\_tools.md)

## G

---

### | Gemini

Google's family of multimodal large language models, including Gemini 1.5, Gemini 2.0, etc.

**See Also:** [Tutorial 22: Model Selection](#) (22\_model\_selection.md)

### | Grounding

Connecting LLM responses to real-world data and facts through tools like web search, databases, and APIs to ensure accuracy.

**See Also:** [Tutorial 11: Built-in Tools & Grounding](#) (11\_built\_in\_tools\_grounding.md)

### | Guardrails

Safety mechanisms and validation rules that prevent agents from performing harmful actions or generating inappropriate content.

**See Also:** [Tutorial 09: Callbacks & Guardrails](#) (09\_callbacks\_guardrails.md)

## L

---

### | Large Language Model (LLM)

AI models trained on vast amounts of text data that can understand and generate human-like text. Examples: Gemini, GPT-4, Claude.

### | Loop Agent

Workflow agent that iteratively refines output through critic/refiner patterns until quality criteria are met.

**See Also:** [Tutorial 07: Loop Agents](#) (07\_loop\_agents.md)

# M

---

## | Memory Service

Persistent storage system for long-term agent memory, enabling agents to recall information across sessions.

**See Also:** [Tutorial 08: State & Memory](#) (08\_state\_memory.md)

## | Model Context Protocol (MCP)

Standardized protocol for tool communication between agents and external services, enabling interoperability.

**See Also:** [Tutorial 16: MCP Integration](#) (16\_mcp\_integration.md)

## | Multi-Agent Systems

Architectures where multiple specialized agents work together to accomplish complex tasks.

**See Also:** [Tutorial 06: Multi-Agent Systems](#) (06\_multi\_agent\_systems.md)

# O

---

## | Observability

The ability to monitor, debug, and understand agent behavior through logging, metrics, and tracing.

**See Also:** [Tutorial 18: Events & Observability](#) (18\_events\_observability.md), [Tutorial 24: Advanced Observability](#) (24\_advanced\_observability.md)

## | OpenAPI Tools

Tools automatically generated from OpenAPI/Swagger specifications, allowing agents to call REST APIs without manual coding.

**See Also:** [Tutorial 03: OpenAPI Tools](#) (03\_openapi\_tools.md)

## | Output Key

Configuration that automatically saves an agent's response to session state for later retrieval.

**See Also:** [Tutorial 08: State & Memory](#) (08\_state\_memory.md)

## P

---

## | Parallel Agent

Workflow agent that executes multiple sub-agents simultaneously for improved performance on independent tasks.

**See Also:** [Tutorial 05: Parallel Processing](#) (05\_parallel\_processing.md)

## | Planners

Advanced reasoning components that help agents break down complex tasks and create execution plans.

**See Also:** [Tutorial 12: Planners & Thinking](#) (12\_planners\_thinking.md)

## | Production Deployment

Strategies for deploying agents to production environments with scalability, reliability, and monitoring.

**See Also:** [Tutorial 23: Production Deployment](#) (23\_production\_deployment.md)

## R

---

### | Runner

ADK component that executes agents, manages state, and coordinates tool calls.

## S

---

### | Sequential Agent

Workflow agent that executes sub-agents in order, where each step depends on the previous step's output.

**See Also:** [Tutorial 04: Sequential Workflows](#) (04\_sequential\_workflows.md)

### | Session State

Key-value storage that persists data within a conversation session but is discarded when the session ends.

**See Also:** [Tutorial 08: State & Memory](#) (08\_state\_memory.md)

### | State Management

System for storing and retrieving data across agent interactions, with different scopes (session, user, app, temp).

**See Also:** [Tutorial 08: State & Memory](#) (08\_state\_memory.md)

### | Streaming

Real-time response generation where the agent sends partial responses as they are generated, rather than waiting for completion.

**See Also:** [Tutorial 14: Streaming & SSE](#) (14\_streaming\_sse.md)

## | Server-Sent Events (SSE)

HTTP standard for real-time communication from server to client, used for streaming agent responses.

**See Also:** [Tutorial 14: Streaming & SSE](#) (14\_streaming\_sse.md)

## T

---

## | Tool Context

Object passed to tool functions containing state, session information, and execution context.

## | Tools

Capabilities that extend agent functionality beyond LLM reasoning. Types include function tools, OpenAPI tools, MCP tools, and built-in tools.

**See Also:** [Tools & Capabilities](#) (tools-capabilities.md)

## V

---

## | Vertex AI

Google Cloud's machine learning platform that provides managed AI services including Gemini models and Agent Engine.

## W

---

## | Workflow Agents

Agents that orchestrate other agents in structured patterns: sequential, parallel, and loop workflows.



**See Also:** [Workflows & Orchestration](#) (workflows-orchestration.md)

## Quick Reference Tables

### Agent Types

Type	Purpose	Example Use Case
<b>LLM Agent</b>	Flexible reasoning and conversation	Customer support, content creation
<b>Sequential Agent</b>	Ordered, dependent steps	Blog writing pipeline
<b>Parallel Agent</b>	Independent concurrent tasks	Research gathering
<b>Loop Agent</b>	Iterative refinement	Code review and improvement

### State Scopes

Prefix	Scope	Persistence	Example
(none)	Current session	SessionService dependent	<code>state['topic']</code>
<code>user:</code>	All user sessions	Persistent	<code>state['user:language']</code>
<code>app:</code>	All users/sessions	Persistent	<code>state['app:settings']</code>
<code>temp:</code>	Current invocation	Never persisted	<code>state['temp:calc']</code>

## Tool Types

Type	Source	Example
<b>Function Tools</b>	Python functions	Custom business logic
<b>OpenAPI Tools</b>	REST API specs	Weather, news APIs
<b>MCP Tools</b>	MCP servers	Filesystem, databases
<b>Built-in Tools</b>	Google ADK	Search, maps, code execution

## Workflow Patterns

Pattern	Execution	Use Case
<b>Sequential</b>	One after another	Assembly line processes
<b>Parallel</b>	All at once	Independent research tasks
<b>Loop</b>	Repeat until criteria met	Quality improvement cycles

## Contributing to the Glossary

This glossary is maintained alongside the ADK tutorials. When new concepts are introduced:

1. Add the term with a clear definition
2. Include "See Also" links to relevant tutorials
3. Update related terms if needed
4. Keep definitions concise but comprehensive

**Last Updated:** October 2025

**ADK Version:** 1.15

---

Generated on 2025-10-19 17:57:40 from glossary.md

Source: Google ADK Training Hub