

目次

- 1. 改訂情報
- 2. はじめに
 - 2.1. Accel Agent とは
 - 2.2. 想定読者と必要な権限
 - 2.3. 本書の読み方
 - 2.4. 業務設計が先
 - 2.5. IM-Copilot の既存機能との関係
- 3. 概念と全体像
 - 3.1. エージェントとは
 - 3.2. AI 活用の適性判断基準
 - 3.3. 部品の棲み分け: ナレッジ・ツール・スキル
 - 3.4. ツールの 3 区分と選び方
 - 3.5. エージェントが動く仕組み
 - 3.6. 開発の流れと本書の対応
- 4. セットアップ
 - 4.1. 前提条件
 - 4.2. 生成AIサービスとモデル
 - 4.3. IM-Juggling プロジェクトの編集
 - 4.4. 生成AI連携ドライバ設定
 - 4.5. ベクトルデータベース接続設定
 - 4.6. トレースの外部送出設定 (OTLP)
 - 4.7. MCPサーバのURLブロックリスト設定
 - 4.8. テナントセットアップ後の設定
- 5. まず動かす (クイックスタート)
 - 5.1. ロールを付与する
 - 5.2. ワークスペースを開く
 - 5.3. AIモデルを登録する
 - 5.4. 最小のエージェントを作る
 - 5.5. テスト実行で確かめる
 - 5.6. 公開して IM-LogicDesigner から呼ぶ
- 6. チュートリアル: 購買申請支援エージェントを業務に組み込む
 - 6.1. このチュートリアルで作るもの
 - 6.2. 業務を設計する
 - 6.3. ナレッジベースを用意する
 - 6.4. ツールを用意する
 - 6.5. エージェントを定義する
 - 6.6. テスト実行で確かめる
 - 6.7. ガードレールを足す
 - 6.8. 精度評価する
 - 6.9. 公開して業務に組み込む
 - 6.10. 運用を始める前に
 - 6.11. 発展
- 7. 本格構築を始める前に
 - 7.1. エージェント定義の画面を開く前に決めること
 - 7.2. 読む順序の目安
- 8. ナレッジベース
 - 8.1. ナレッジベースで確かめること
 - 8.1.1. 製品がやること・利用者がやること
 - 8.1.2. 取り込めるファイル
 - 8.1.3. 精度は検索と推論の掛け算で決まる
 - 8.1.4. 画面を開く
 - 8.2. ナレッジベースを設計する
 - 8.2.1. 粒度を決める
 - 8.2.2. 2 つの説明欄を書き分ける
 - 8.3. アイテムを取り込む
 - 8.3.1. 詳細設定で指定できること
 - 8.3.2. 登録後に確認すること
 - 8.4. チャンク分割を設定する
 - 8.4.1. 再ベクトル化のやり方
 - 8.5. メタデータ項目を設計する
 - 8.5.1. メタデータが使われる場面

- 8.5.2. ビルトイン項目とカスタム項目
 - 8.6. 有効期間で検索対象を管理する
 - 8.7. 埋め込みモデルの変更とナレッジベースの削除
 - 8.8. インデックス状況と利用箇所を確認する
 - 8.9. インポート・エクスポート
 - 8.10. 精度が出ないときの切り分け
 - 8.11. 困ったときの確認
- 9. ツール
 - 9.1. ツールで確かめること
 - 9.1.1. LLMに公開されるツール名の一意化
 - 9.1.2. 画面を開く
 - 9.2. 組み込みツールを割り当てる
 - 9.3. ロジックツールを作る
 - 9.4. MCPサーバに接続する
 - 9.4.1. OAuth の認証プロバイダを登録する
 - 9.5. MCPのセキュリティに注意する
 - 9.6. 副作用のあるツールの扱い方
 - 9.7. 困ったときの確認
- 10. スキル
 - 10.1. スキルで確かめること
 - 10.1.1. 画面を開く
 - 10.2. スキルを構成する要素
 - 10.3. スキルを作る
 - 10.4. スキルが使われる仕組み
 - 10.5. エージェントへの割り当てと利用箇所の確認
 - 10.6. インポート・エクスポート
 - 10.7. 困ったときの確認
- 11. エージェント定義
 - 11.1. 画面を開く・全体の流れ
 - 11.2. 基本情報・モデル設定を決める
 - 11.2.1. 基本情報
 - 11.2.2. モデル設定
 - 11.3. 指示を書く
 - 11.4. 入出力値を設計する
 - 11.5. ナレッジベース・スキル・ツール・ガードレールを割り当てる
 - 11.5.1. ナレッジベース
 - 11.5.2. スキル
 - 11.5.3. ツール
 - 11.5.4. ガードレール
 - 11.6. 実行制御を設定する
 - 11.7. テスト実行で確認する
 - 11.8. バージョンを管理し、公開して改善する
 - 11.9. 困ったときの確認
- 12. ガードレール
 - 12.1. ガードレールで確かめること
 - 12.2. 検査位置を選ぶ
 - 12.3. 動作を選ぶ
 - 12.4. 判定スクリプトを書く
 - 12.4.1. runの引数
 - 12.4.2. 返り値の書き方
 - 12.4.3. 書き方の例
 - 12.4.4. 失敗したときの扱い
 - 12.5. 逐次応答の検査設定
 - 12.6. 保存前にテストする
 - 12.7. 発動状況を確認する
 - 12.8. 困ったときの確認
- 13. 精度評価
 - 13.1. 評価で確かめること
 - 13.1.1. スコアと業務上の合否
 - 13.1.2. 評価種別の選び方
 - 13.1.3. 評価に使う情報の単位
 - 13.2. 評価を始める準備
 - 13.2.1. 画面を開く
 - 13.2.2. 必要な権限

- 13.2.3. 対象とモデルの準備
 - 13.2.4. 実行先の確認
 - 13.3. テストケースの作り方
 - 13.3.1. 通常の質問と失敗しやすい質問
 - 13.3.2. RAGの質問と模範回答
 - 13.3.3. エージェントの期待回答と判定条件
 - 13.3.4. 自動生成したケースの確認
 - 13.4. 新しい評価計画を作る
 - 13.4.1. 「対象と種別」を設定する
 - 13.4.2. 「データセット」を準備する
 - 13.4.3. 「採点条件」を設定する
 - 13.4.4. 「確認」で保存または実行する
 - 13.5. データセットを確認・修正する
 - 13.6. 実行状況とやり直し
 - 13.7. 結果の読み方
 - 13.7.1. 件数とエラーを先に確認する
 - 13.7.2. RAGの指標から調べる箇所
 - 13.7.3. エージェントの指標から調べる箇所
 - 13.7.4. スコア平均と前回比較
 - 13.7.5. 時間と費用
 - 13.8. 改善して同じ条件で測る
 - 13.9. レポートの保存
 - 13.10. 困ったときの確認
 - 13.11. 設定値
 - 13.11.1. 採点条件と対象バージョン
 - 13.11.2. データセットの共通制約
 - 13.11.3. RAGの入力形式 (CSV・TSV・Excel)
 - 13.11.4. エージェント評価の入力形式 (JSON)
 - 13.11.5. 呼び出し条件と証拠
 - 13.11.6. 実行のやり直しに必要な条件
 - 13.11.7. RAGの対象外条件
 - 13.11.8. エージェントの対象外条件
 - 13.11.9. 履歴比較の条件
 - 13.11.10. 改善提案・権限区分
- 14. 設定リファレンス
 - 14.1. AIモデル管理
 - 14.2. トレースの外部送出設定ファイル
- 15. 業務に組み込む
 - 15.1. エージェントの呼び出し方
 - 15.2. IM-LogicDesigner の「エージェント実行」タスク
 - 15.3. IM-BPM から呼ぶ
 - 15.4. Java から呼ぶ
 - 15.5. バージョンの更新と進行中案件
- 16. 実行を追う (トレース)
 - 16.1. 何が見える画面か
 - 16.2. トレース一覧・詳細の読み方
 - 16.3. 評価結果からトレースへ
 - 16.4. コストの見方
 - 16.5. 外部の監視基盤で追う
 - 16.6. 保持期間と閲覧権限
 - 16.7. 読み違いやすい点
- 17. インポート／エクスポート
 - 17.1. インポート・エクスポートできる対象
 - 17.2. 共通の進め方と事故への備え
 - 17.3. インポート先の前提とインポートの順番
 - 17.4. エクスポートの注意点
 - 17.5. インポートで何が起きるか
 - 17.6. インポート後に確かめること
- 18. 運用ダッシュボード
 - 18.1. 何が見える画面か
 - 18.2. KPI の 4 枚
 - 18.3. 推移グラフ
 - 18.4. 内訳と一覧
 - 18.4.1. エラー種別内訳

- 18.4.2. スパン種別別 平均所要時間
- 18.4.3. ツール別利用状況
- 18.4.4. ナレッジ検索ゼロヒット
- 18.4.5. 直近エラー一覧
- 18.4.6. エージェント別ランキング
- 18.5. トレースへの掘り下げ
- 18.6. 読み換えやすい点
- 19. 経営ダッシュボード
 - 19.1. 何が見える画面か
 - 19.2. KPI の 5 枚
 - 19.3. 推移と内訳
 - 19.4. 金額を読むときの前提
 - 19.5. グループ解決設定
 - 19.6. 読み換えやすい点
- 20. 監査ログ
 - 20.1. 何が記録されるか
 - 20.2. 記録されないもの
 - 20.3. 変更内容の粒度
 - 20.4. 検索と CSV
 - 20.5. トレースとの行き来
 - 20.6. 保持と増加
- 21. ジョブの運用
 - 21.1. ジョブネットの作成と有効化
 - 21.2. ジョブごとの設定
 - 21.3. 運用の注意
- 22. ロールと権限
 - 22.1. 同梱ロールが許すもの
 - 22.2. 参照のみのロールの作り方
 - 22.2.1. 役割ごとの設定例
 - 22.3. 参照だけでは足りない場面
 - 22.4. 権限と画面の見え方
 - 22.5. 見える範囲の広さ

改訂情報

変更年月日	変更内容
2026-10-01	初版

はじめに

項目

- [Accel Agent とは](#)
- [想定読者と必要な権限](#)
- [本書の読み方](#)
- [業務設計が先](#)
- [IM-Copilot の既存機能との関係](#)

Accel Agent とは

Accel Agent は、IM-Copilot の一機能として提供する、業務エージェントの作成・実行・評価機能です。ナレッジ・ツール・スキルを組み合わせでエージェントを定義し、テスト実行で応答を確かめてから公開し、IM-LogicDesigner や IM-BPM から呼び出して業務に組み込みます。

Accel Agent には次の特徴があります。

- エージェント定義・ナレッジベース・ツール（組み込みツール／ロジックツール／MCP）・スキル・AIモデル管理・精度評価・トレース／ダッシュボード／監査ログという一連の機能群で、エージェントの作成から運用までを一つの製品でまかなえます。
- エージェントの分岐処理をあらかじめフローとして設計する必要がなく、LLM が状況に応じてツールを呼び出す順番を判断します。そのぶん、期待どおりに動くかどうかはテスト実行と精度評価で確かめる必要があります。
- IM-Copilot が既に提供しているドライバ・アクション・ベクトルデータベース操作機能の上に構築されています。生成AI連携ドライバ設定やベクトルデータベース接続設定は、既存の IM-Copilot 機能と共有します（「[セットアップ](#)」）。

Accel Agent のエージェント定義は、IM-Copilot の既存のアシスタント定義とは別の機能です。両者は画面もデータも独立しており、アシスタント定義の設定を流用することはできません。

想定読者と必要な権限

Accel Agent は、環境を用意する管理者、エージェントを作る開発者、エージェントを利用する運用管理者・業務管理者・経営層、そして監査担当という異なる読者が関わります。読者ごとに必要な権限と、読むべき章は次のとおりです。

読者	必要な権限	読む章
環境を用意する管理者	テナント管理者	「 セットアップ 」、「 インポート／エクスポート 」、「 ジョブの運用 」、「 ロールと権限 」
エージェントを作る開発者	Accel Agent 管理者ロール	「 概念と全体像 」、「 まず動かす（クイックスタート） 」～「 ガードレール 」（着手前に「 ベクトルデータベース接続設定 」と、「 ジョブネットの作成と有効化 」を済ませておきます）
運用管理者・業務管理者・経営層	参照のみのロールで可	「 運用ダッシュボード 」、「 経営ダッシュボード 」
監査担当	参照のみのロールで可	「 監査ログ 」

エージェントを作るには、Accel Agent 管理者ロール（またはテナント管理者）が必要です。ダッシュボードと監査ログの閲覧は、参照のみのロールで足りません。Accel Agent 管理者ロールは、Accel Agent の各機能に加えて、ロジックフローの編集と OAuth プロバイダ設定も許可します（「[ロールと権限](#)」）。IM-LogicDesigner の基本操作は、本書では既知として扱います。

本書の読み方

本書には、目的別に次の入口があります。

- 環境を用意したい場合は、「[セットアップ](#)」から読みます。
- まず動かして雰囲気をつかみたい場合は、「[まず動かす（クイックスタート）](#)」から読みます。
- 業務に組み込むところまでを一度通して試したい場合は、「[チュートリアル: 購買申請支援エージェントを業務に組み込む](#)」から読みます。
- 本格的にエージェントを作り込みたい場合は、「[本格構築を始める前に](#)」から読みます。
- 個々の設定項目の意味を調べたい場合は、「[設定リファレンス](#)」を辞書として引きます。
- 別の環境へ定義を移したい場合は、「[インポート／エクスポート](#)」から読みます。

ナレッジベース・ツール・ガードレールは、「[チュートリアル: 購買申請支援エージェントを業務に組み込む](#)」で一通り触れたうえで、「[ナレッジベース](#)」・「[ツール](#)」・「[ガードレール](#)」で本格的に扱います。精度評価は「[精度評価](#)」で扱います。スキルは「[スキル](#)」が最初の入口です。

業務設計が先

Accel Agent を使い始める前に、どの領域をエージェントに任せ、どの領域を IM-LogicDesigner や IM-BPM のフロー制御に残すかを決めておく必要があります。この判断は、「[概念と全体像](#)」の AI 活用の適性判断基準を参考にしてください。

IM-Copilot の既存機能との関係

Accel Agent 利用ガイド — Accel Agent 利用ガイド 初版 2026-10-01

生成AI連携ドライバ設定とベクトルデータベース接続設定は、IM-Copilot の既存機能と共有します。参照先は「[セットアップ](#)」で示します。

AIモデル管理は、IM-Copilot の「サポートモデル」（デフォルトモデルとサポートモデルの二層構造）の枠組みを持ちません。Accel Agent で任意に登録したモデルのうち、動作を保証するものは、「[生成AIサービスとモデル](#)」を参照してください。

IM-LogicDesigner のタスクカテゴリ「IM-Copilot」には、既存の IM-Copilot タスクと Accel Agent のタスクが同居します。「[業務に組み込む](#)」で扱う「エージェント実行」タスクなどは、このカテゴリから選びます。

概念と全体像

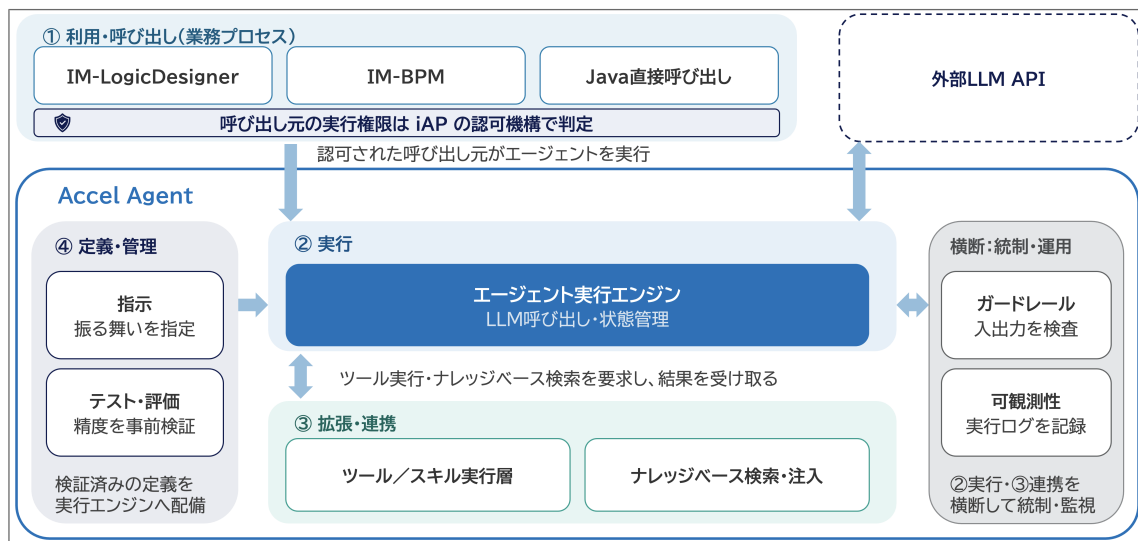
項目

- エージェントとは
- AI 活用の適性判断基準
- 部品の棲み分け: ナレッジ・ツール・スキル
- ツールの 3 区分と選び方
- エージェントが動く仕組み
- 開発の流れと本書の対応

エージェントとは

Accel Agent におけるエージェントとは、LLM・ツール・ナレッジ・制御を組み合わせた実行単位です。IM-LogicDesignerのフローやIM-BPMのプロセスがあらかじめ分岐を設計しておくのに対し、エージェントは分岐を事前に設計しません。次に何をすべきかは、LLM がそのつど判断します。この判断に最終責任を持つのは人であり、エージェントの提案をそのまま自動実行するかどうかは、呼び出し側の業務設計で決めます。

エージェントは単体で完結するものではなく、IM-LogicDesigner や IM-BPM の中で呼ばれるタスク型として使うのが中心的な使い方です。呼び出し元を絞る「用途」には、ロジックフロータスクと BPMタスクがあります。



図：Accel Agent の機能構成。呼び出し元から実行エンジンを呼び、ツール・ナレッジとの連携、定義・管理、ガードレールと可観測性による統制が実行を支える。

自分が扱おうとしている課題がエージェント向きか、それとも通常のロジックフローで済むかは、「[AI 活用の適性判断基準](#)」の判断基準で見極めます。

AI 活用の適性判断基準

すべての業務課題が、AI による自動化に向いているわけではありません。特に次のような業務では、AI に判断や実行を全面的に任せることには注意が必要です。

- 判断結果について、厳密かつ客観的な根拠の提示が求められる
- 誤りが許容できない、または誤りを人やシステムで検証することが難しい
- 判断基準を明確にできず、個々の状況に応じた高度な判断が求められる

一方、多少の誤りを許容または検証でき、AI に任せる範囲を明確にできる業務は、AI を活用しやすい領域です。特に、文章の理解や情報の抽出・整理、分類、要約、下書き作成などは、生成AIの特性を活かしやすい業務です。

AI による自動化に向かない業務であっても、AI の活用そのものを諦める必要はありません。たとえば、AI に下書きや候補だけを作らせ、最終的な判断や確定を人が行う方法があります。

AI に任せる度合いにも段階があります。AI が候補を提示する「提案型」、人が確認して初めて実行する「確認型」、人の確認なしに実行まで進む「完全自動型」です。どの段階を選ぶかによって、AI に任せる範囲と人の関与点が変わります。

Accel Agent 自体は、どの段階を選択するかを強制しません。業務の特性やリスクに応じて、呼び出し元のフローで設計します。

部品の棲み分け: ナレッジ・ツール・スキル

エージェントに情報や手順を持たせる方法には、ナレッジ、ツール、スキルの 3 種類があります。この 3 つは役割が異なるため、どの情報をどれで持たせるかは、実装に入る前に決めておく必要があります。

- **ナレッジ**: 事前にインデックス化しておき、実行時には意味検索で取り出す情報です。社内規程やマスターデータのように、頻繁には変わらず、文書として蓄積されている情報が向きます。
- **ツール**: 実行時に正本のシステムへ問い合わせる、または正本の状態を変える処理です。在庫の最新値のように、その場で取得しないと古くなってしまふ情報や、案件の申請・承認のように状態を変える操作は、ナレッジではなくツールで扱います。

- **スキル:** LLM に読ませる手順書です。ツールやナレッジのような実行部品ではなく、申請書の書き方のような「進め方の知識」を LLM に渡すために使います。

例えば、社内の購買承認ポリシーは頻繁に変わらない規程なのでナレッジに、在庫の最新数量は実行のたびに変わるのでツールに、申請書の書き方の手順はスキルに、というように切り分けます。

ツールの 3 区分と選び方

ツールには、組み込み、ロジック、MCP の 3 区分があります。

- **組み込みツール:** IM-Workflow など、他製品が標準で提供するツールです。供給元の製品を導入していない環境では選択肢に現れません。
- **ロジックツール:** エージェントを作る開発者が、IM-LogicDesigner のロジックフローとして定義するツールです。業務固有で、変更が多い処理に向きます。
- **MCP ツール:** MCP プロトコル経由で外部サービスに接続するツールです。intra-mart Accel Platform の権限コンテキストは引き継がれないため、外部サービス側でどの権限として動くかを別途設計する必要があります。

まず組み込みツールで足りるかどうかを確認し、足りない場合にロジックツールか MCP のどちらで補うかを選ぶ、という順序で検討します。詳細な選び方は「[ツール](#)」で扱います。

エージェントが動く仕組み

エージェントの実行は、ループの繰り返しで進みます。1 回のループは、LLM 呼び出し 1 回とそれに続くツール実行から成ります。LLM は、割り当てられたツールやナレッジの検索結果を見ながら、次に何をすべきかをループのたびに判断します。

ツールの実行が失敗しても、エージェント全体の実行は止まりません。エラーの内容が LLM に返され、LLM は次のループでその内容を踏まえて判断を続けます。

実行の進め方には、シンプル、Agentic RAG、Simple RAG、Plan & Action、Deep Research といった戦略があります。戦略の選び方は「[実行制御を設定する](#)」で扱い、エージェント定義の各項目の詳細な設定値は「[エージェント定義](#)」で扱います。

エージェントのバージョンには、公開・非公開の状態があります。どの状態のバージョンを実行できるかはエージェント側では制御せず、呼び出し元の機能が決めます。例えば、IM-LogicDesigner の「エージェント実行」タスクは公開中のバージョンだけを呼び出せますが、テスト実行や精度評価では非公開のバージョンも実行できます（「[IM-LogicDesigner の「エージェント実行」タスク](#)」）。

開発の流れと本書の対応

Accel Agent を使ったエージェント開発は、課題認識、構想、開発（ナレッジ・ツール・スキル・定義の作成）、精度評価、業務への組み込み、別環境へのインポート、運用改善という流れで進みます。

「[チュートリアル: 購買申請支援エージェントを業務に組み込む](#)」は、この流れを一つの題材で一通りたどる構成です。個々の工程を本格的に扱う「[本格構築を始める前に](#)」以降へ進む前に、まず「[チュートリアル: 購買申請支援エージェントを業務に組み込む](#)」で全体の流れをつかむことができます。

セットアップ

本章では、Accel Agent を使い始めるための環境構築（モジュールの追加、設定ファイルの編集）と、テナント環境セットアップ、そしてその直後に管理者が行う設定までを扱います。IM-Copilot の既存機能と共有する手順は、参照先のドキュメントに従って設定します。

項目

- 前提条件
- 生成AIサービスとモデル
- IM-Juggling プロジェクトの編集
- 生成AI連携ドライバ設定
- ベクトルデータベース接続設定
- トレースの外部送出設定（OTLP）
- MCPサーバのURLブロックリスト設定
- テナントセットアップ後の設定

前提条件

Accel Agent は、intra-mart Accel Platform 2026 Autumn 以降で利用できます。利用できるライセンス種別・エディションの詳細は、[リリースノート](#)の「システム要件」を参照してください。

生成AIサービスとモデル

エージェントにはテキスト生成モデルが、ナレッジベースには埋め込みモデルが必要です。

2026 Autumnリリース時点の動作保証モデル（テキスト生成）は、次のとおりです。

動作保証モデル（2026 Autumn）

生成AIサービス	テキスト生成モデル
OpenAI／Azure OpenAI	<code>gpt-5.1</code> 、 <code>gpt-5.2</code> 、 <code>gpt-5.4</code> 、 <code>gpt-5.5</code> 、 <code>gpt-5.6-sol</code>
Amazon Bedrock	Claude Sonnet 4.5、Claude Sonnet 4.6、Claude Sonnet 5

`gpt-5.6-sol` でツールを使用する場合は、AIモデル管理で「推論を無効にする」を設定してください。このモデルでは、ツール呼び出しと推論を併用できません。

埋め込みモデルは、この動作保証モデルの一覧に含まれません。

Accel Agent では、一覧にないモデルも AIモデル管理に任意で登録・利用できますが、動作保証モデル以外については動作・性能を保証しません。利用前に動作を確認してください。

コラム

AIモデル管理のドライバ種別には、Ollama や vLLM などのローカル LLM に接続するための「OpenAI互換サービス」もありますが、上表の動作保証モデルには含まれません。IM-Copilot 利用ガイドの「[OpenAI API 互換を使用する場合](#)」に動作検証済みのサービスとモデルが記載されていますが、これは IM-Copilot の機能での検証であり、Accel Agent での動作を保証するものではありません。利用する場合は、事前に動作を確認してください。

IM-Juggling プロジェクトの編集

Accel Agent を利用するには、IM-Juggling プロジェクトに IM-Copilot モジュールを追加します。Accel Agent は IM-Copilot モジュールに含まれており、単独のモジュールはありません。

コラム

- Amazon Bedrock を利用する場合は、IM-Copilot Amazon Bedrock ドライバモジュールもあわせて選択してください。
- 組み込みツールを利用する場合は、その供給元となる製品（IM-Workflow など）のモジュールが必要です。
- BPMタスクとして利用する場合は、IM-BPM のモジュールが必要です。

生成AI連携ドライバ設定

Accel Agent が利用するドライバ種別ごとに、生成AI連携ドライバ設定で driver エントリを設定します。手順は IM-Copilot 利用ガイドの[セットアップ（生成AI）](#)を参照してください。

コラム

AIモデル管理では、モデルへの接続先や API キーを保持しません。接続情報はドライバ設定側が持ちます。また、IM-Copilot の生成AI連携ドライバ設定にあるデフォルトモデルの指定は、Accel Agent では使用しません。Accel Agent で使うモデルは、「[AIモデル管理](#)」で個別に登録します。

ベクトルデータベース接続設定

Accel Agent のナレッジベースを利用するには、テナント環境セットアップで**ベクトルデータベース接続情報**を設定する必要があります。この設定は IM-Copilot の既存機能と共通です。

トレースの外部送出設定（OTLP）

トレースを外部の監視基盤へ送出する設定です。設定は任意で、既定では無効です。

設定は次の 2 段階で行います。

1. 設定ファイル `im-copilot-otlp-export-config.xml` で、外部送出そのものを有効にします。
2. テナント環境セットアップの「テレメトリの外部送出設定」で、「送出先URL」「認証方式」（「なし」または「APIキー」）「認証情報」「追加HTTPヘッダ」を設定します。



コラム

設定ファイル側が無効（`enabled=false`）のままだと、手順 2 のテナント環境セットアップ側で値を設定しても送出されません。この関係は画面上にも案内が表示されます。

設定しない場合の挙動

外部送出を設定しない場合、トレースは Accel Agent の画面内（「**実行を追う（トレース）**」）でのみ確認できます。外部の監視基盤で横断的に見ることはできません。

関連する制限

外部送出の対象はトレース（スパン）のみです。メトリクス・ログは対象に含まれません。送出される内容にプロンプトや応答の本文は含まれず、属性のみが送られます（「**外部の監視基盤で追う**」）。

設定ファイルの主な項目は、バッチ関連（キュー長、送出間隔、タイムアウト）、接続関連（タイムアウト、最初の送出を含めた試行回数の上限 2~5 回、バックオフ）、プロキシです。個々の項目は「**トレースの外部送出設定ファイル**」を参照してください。

MCPサーバのURLブロックリスト設定

MCP ツールを利用する場合にのみ関係する設定です。設定ファイル `mcp-url-blocklist-config.xml`（画面上の表示名は「Accel Agent MCPサーバURLブロックリスト設定」）で、MCP サーバとして接続できる URL を制限します。



注意

出荷時の既定値では、プライベートアドレス帯（IPv4 は 6 レンジ、IPv6 は 5 レンジ）への接続を拒否します。そのため、**プライベートアドレス帯に置かれた MCP サーバは、既定の設定のままでは登録も実行もできません**。このような MCP サーバを利用する場合は、該当するアドレスレンジを設定ファイルから外してください。

URL パターンを前方一致と で指定して、特定のドメインだけを拒否する使い方もできます。

設定しない場合の挙動

設定ファイルが存在しない場合は、URL の制限なしに全て許可されます。既定のブロックリストを使いたい場合は、ファイルを配置したままにしてください。

関連する制限

- `http` / `https` 以外のスキームは、設定に関わらず常に拒否されます。
- 判定は MCP サーバの登録時だけでなく、ツールを実行するたびにも行われます（DNS リバインディング対策）。
- 設定ファイルの編集は、保存すると即座に反映されます。

テナントセットアップ後の設定

テナント環境セットアップの完了後、Accel Agent を使い始めるまでに、次の 3 つの設定が必要です。

1. ロールの付与（管理者の作業。「**ロールと権限**」）
2. ジョブネットの作成と有効化（管理者の作業。「**ジョブの運用**」）。ジョブネットは製品に同梱されないため、作成して有効化しないと、ナレッジベースのベクトル化ジョブが動きません。
3. AIモデルの登録（ロールを付与された開発者の作業。「**AIモデル管理**」）

Accel Agent 利用ガイド — Accel Agent 利用ガイド 初版 2026-10-01

まず動かす（クイックスタート）

本章は、「**セットアップ**」の生成AI連携ドライバ設定が完了していることを前提に、最小構成のエージェントを作って動かすところまでを扱います。ロールの付与とAIモデルの登録は本章の中で行うため、事前の準備は必要ありません。ナレッジベースは使わないため、ベクトルデータベース接続設定とジョブネットの設定も、本章の範囲では不要です。

項目

- ロールを付与する
- ワークスペースを開く
- AIモデルを登録する
- 最小のエージェントを作る
- テスト実行で確かめる
- 公開して IM-LogicDesigner から呼ぶ

ロールを付与する

テナント管理者が、エージェントを作成するユーザに Accel Agent 管理者ロールを付与します。手順は テナント管理者操作ガイドの[認可を設定する](#)を参照してください。テナント管理者自身のアカウントで、ロールを付与せずにそのまま進めることもできます。

コラム

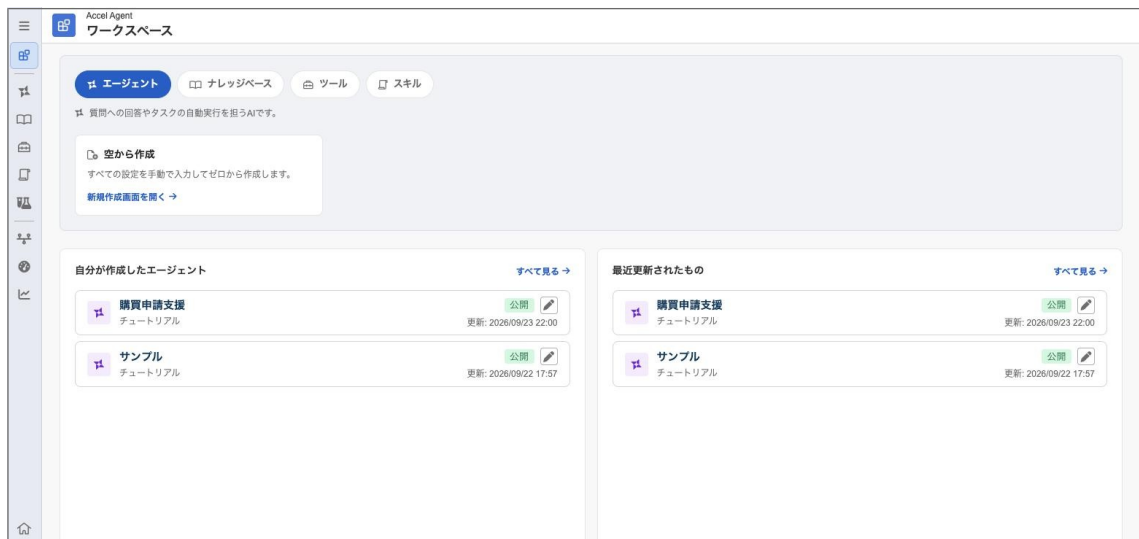
Accel Agent 管理者ロールを付与すると、サイトマップの「Copilot」配下に「AIモデル」と「Accel Agent」の項目が現れます。このロールは、Accel Agent の各機能に加えて、ロジックフローの編集と OAuth プロバイダ設定（MCP の OAuth 認証で使用）も許可します。

ワークスペースを開く

サイトマップから「Copilot」→「Accel Agent」→「ワークスペース」を開きます。ワークスペースは Accel Agent の入口となる画面で、エージェント・ナレッジベース・ツール・スキルを作成するタブと、それぞれの部品の一言説明、作成経路へのカード、そして自分が作成したエージェントや最近更新されたものの一覧が並びます。以降の作成作業は、この画面から始めます。



図：サイトマップの「Copilot」→「Accel Agent」。ロールを付与すると、この配下に各管理画面が現れる。



図：ワークスペース画面。エージェント・ナレッジベース・ツール・スキルの作成タブと、自分が作成したものの一覧が並ぶ。

AIモデルを登録する

サイトマップから「Copilot」→「AIモデル」→「AIモデル管理」を開き、ID（AIモデルID）・ドライバ種別・プロバイダモデルID・モデル種別（「テキスト生成」を選択）・コンテキストウィンドウを入力して 1 件登録します。「[チュートリアル: 購買申請支援エージェントを業務に組み込む](#)」まで進む場合は、配布したエージェント定義の参照先に合わせ、AIモデルID を `gpt-5.4` にします。OpenAI を利用する場合は、プロバイダモデルID も `gpt-5.4` にします。

コラム

ここで選ぶモデルは、「[セットアップ](#)」で設定した生成AI連携ドライバから接続できるものである必要があります。パラメータの仕様や単価は、登録後に編集できます。詳細は「[AIモデル管理](#)」を参照してください。

AIモデル編集画面のスクリーンショット。画面例のIDは `gpt-5-mini`。基本情報、プロバイダモデル、コンテキストウィンドウなどの設定項目が確認できる。

図：AIモデル編集画面（画面例の ID は `gpt-5-mini`）。購買申請支援のチュートリアルでは、本文のとおり `gpt-5.4` を登録する。

最小のエージェントを作る

ワークスペースの「空から作成」から「新規作成画面を開く」を選びます（エージェント管理の一覧画面からも作成できます）。保存に必須の項目は、エージェント ID（自動採番）、エージェント名、カテゴリ、AIモデルの 4 つです。左側のメニューで「指示」をクリックし、表示された指示の入力欄に次の文を入力します。「基本情報」の「説明」欄には入力しません。

指示の例

あなたは社内問い合わせ対応の一次窓口です。ユーザからの質問に、簡潔かつ丁寧な日本語で回答してください。回答に自信が持てない場合は、正直に「わかりません」と答え、担当部署への確認を促してください。

エージェント編集画面の「指示」タブのスクリーンショット。システムプロンプト欄に指示の例を入力した状態が確認できる。

図：エージェント編集画面の「指示」。システムプロンプト欄に指示の例を入力した状態。

テスト実行で確かめる

編集画面の右側の領域から、保存せずにエージェントを実行できます。AIモデルが選択されていない状態では実行できません。テキストエリアに質問を入力して実行します。

テスト実行の入力例

経費精算の申請方法を教えてください。

実行結果には、応答本文のほか、トークン数と所要時間が表示されます。テスト実行は、公開後の本番実行とは区別して記録されます。

基本情報

エージェントID *
8t5lgn42lsyl41t

エージェント名 *
サンプル

説明
どのような用途のエージェントかを記述します。エージェントを選ぶときの判断材料になります。
例: 経費精算の申請内容をチェックし、不備を指摘します

カテゴリ *
チュートリアル

用途
このエージェントを利用する機能を選択してください。選択されていないと、利用先で選択できません。
☒ **ロジックフロータスク**
IM-LogicDesigner のフロー内タスクとしてエージェントを実行します。

モデル設定

AIモデル選択
AIモデル *

テスト実行

6. 上長の承認ルートに申請を送る (システムで自動振り分けされることが多い)
7. 経理部での確認後、差戻しがあれば修正して再申請、問題なければ支払処理される
8. 支払スケジュール (締め日・支払日) に使い振込される

注意点:
- 領収書の原本提出が必要 (電子データで可) は会社ルールによります。
- 外貨や立替経費、交際費などは別ルールや上長の許可が必要な場合があります。
- 月締めや申請期限があるので、規程を確認してください。

確認先・ヘルプ:
- 社内の「経費規程」または「経理部 (経費担当)」に問い合わせてください。
- どのシステムを使っているか教えていただければ、より具体的な手順を案内します。

入力 88 / 出力 1065 トークン・15669ms
トレースID g5M4c3M8-f7b8-47ab-8f3b-c8bc74a99627 2

メッセージを入力... (Enter で送信)

図：テスト実行の結果。応答本文とともに、トークン数と所要時間が表示される。

公開して IM-LogicDesigner から呼ぶ

エージェント定義の「基本情報」内の「用途」で「ロジックフロータスク」を ON にします。設定したら、画面右上の「保存」をクリックします。続いて、その隣の「公開」をクリックします。新規作成の場合、「公開」は保存するまで表示されません。ここまでが Accel Agent の画面での作業です。

コラム

用途を選んでいないと、IM-LogicDesigner でタスクを配置するときの一覧にこのエージェントが表示されません。また、公開していないエージェントは、IM-LogicDesigner のフローから呼び出せません。

基本情報

エージェントID *
8t5lgn42lsyl41t

エージェント名 *
サンプル

説明
どのような用途のエージェントかを記述します。エージェントを選ぶときの判断材料になります。
例: 経費精算の申請内容をチェックし、不備を指摘します

カテゴリ *
チュートリアル

用途
このエージェントを利用する機能を選択してください。選択されていないと、利用先で選択できません。
☒ **ロジックフロータスク**
IM-LogicDesigner のフロー内タスクとしてエージェントを実行します。

モデル設定

AIモデル選択
AIモデル *

テスト実行

6. 上長の承認ルートに申請を送る (システムで自動振り分けされることが多い)
7. 経理部での確認後、差戻しがあれば修正して再申請、問題なければ支払処理される
8. 支払スケジュール (締め日・支払日) に使い振込される

注意点:
- 領収書の原本提出が必要 (電子データで可) は会社ルールによります。
- 外貨や立替経費、交際費などは別ルールや上長の許可が必要な場合があります。
- 月締めや申請期限があるので、規程を確認してください。

確認先・ヘルプ:
- 社内の「経費規程」または「経理部 (経費担当)」に問い合わせてください。
- どのシステムを使っているか教えていただければ、より具体的な手順を案内します。

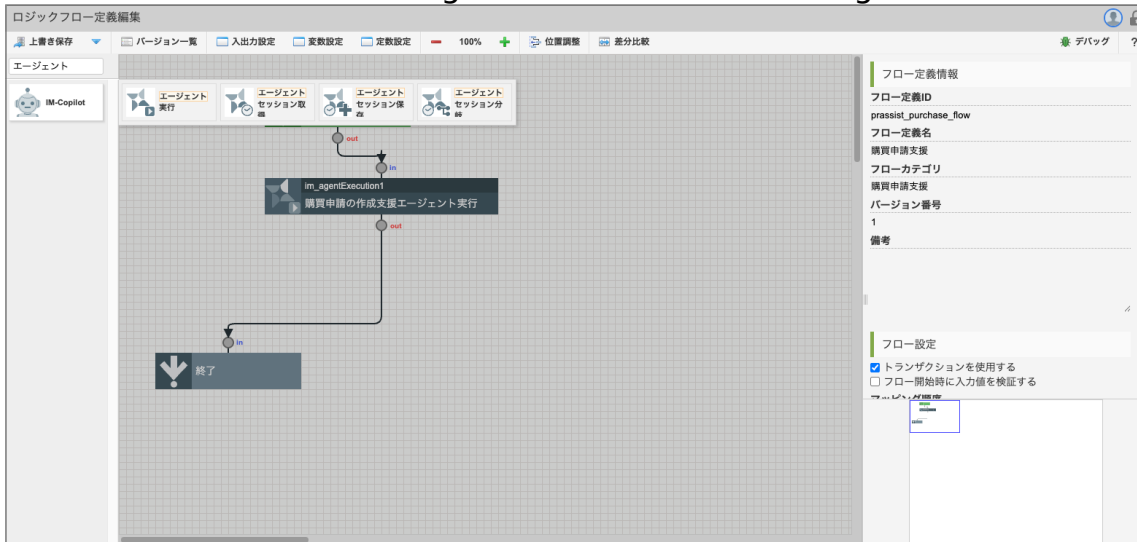
入力 88 / 出力 1065 トークン・15669ms
トレースID g5M4c3M8-f7b8-47ab-8f3b-c8bc74a99627 2

メッセージを入力... (Enter で送信)

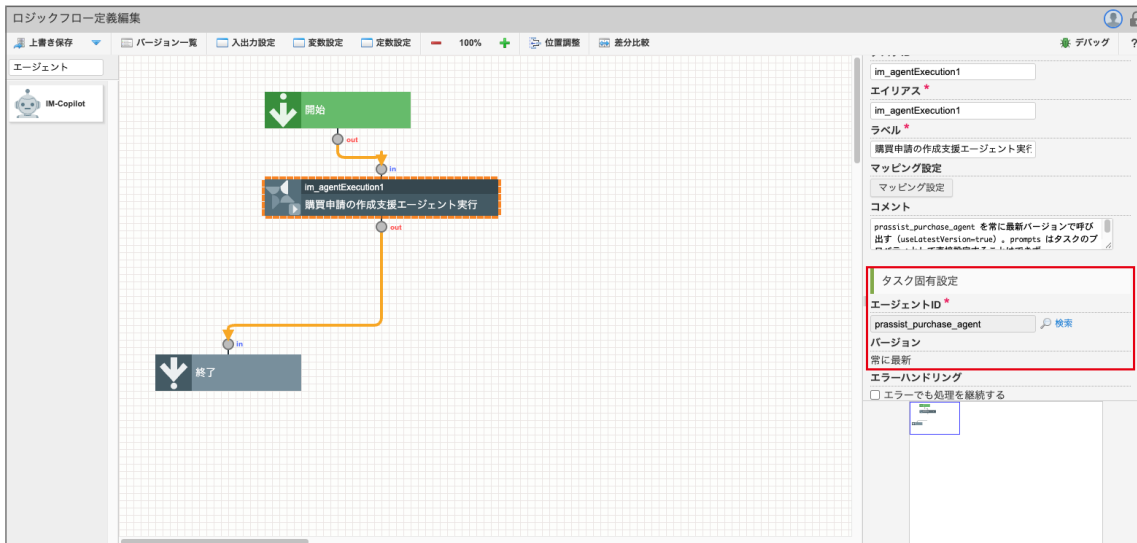
図：用途「ロジックフロータスク」を ON にした状態。この後、保存・公開の操作に進む。

ここから先は IM-LogicDesigner の画面での作業です。IM-LogicDesigner を開き、動作確認用のフロー定義を新規に作成するか、既存のフロー定義を開いてフローエディタ画面に入ります。

- パレットの「IM-Copilot」カテゴリから「エージェント実行」タスクをキャンバスに配置します。
- タスクのプロパティの「タスク固有設定」で、エージェントID の検索アイコンからエージェントを選び、バージョンに「常に最新のバージョンを利用する」を指定します。
- タスクのマッピング設定で、**prompts** の要素に、**text** として、先ほどテスト実行で使った質問（「経費精算の申請方法を教えてください」など）を指定します。



図：IM-LogicDesignerのパレットに現れる「エージェント実行」タスクと、配置後のキャンバス上のフロー。



図：タスクのプロパティ。「タスク固有設定」（赤枠）で、エージェントID とバージョン（「常に最新」）を指定する。

開始タスクから終了タスクまでが線でつながった、フローとして成立する状態にしてから保存します。保存したフロー定義をデバッグ実行すると、エージェントが呼び出され、`result.message` に応答が入って返ってくることを確認できます。

注意

エージェントを公開しないまま呼び出すと、実行時に「公開されているバージョンが存在しません」というエラーが発生します。呼び出す前に、必ず公開したことを確認してください。

Accel Agent 利用ガイド — Accel Agent 利用ガイド 初版 2026-10-01

チュートリアル: 購買申請支援エージェントを業務に組み込む

購買申請の画面は、多くの現場ですでに動いています。件名や購入先、明細を入力し、勘定科目を選び、実施理由や物品選定理由を書いて申請する画面です。勘定科目と理由文は申請者が手作業で埋めるため、見積書の品目と科目の対応を取り違えたり、記入に手間取ったりします。こうした記入の誤りは、承認の過程で差戻しという形で申請者に戻ってくることも珍しくないでしょう。

図：既存の購買申請画面。勘定科目、実施理由・背景、物品選定理由、購入先選定理由を申請者が手作業で埋める。

申請者が手を止める箇所はだいたい決まっています。見積書のどの数字をどの勘定科目に振り分ければよいか、そして、その振り分けが社内の承認ポリシーに照らして通るのかどうかです。前者は経理の知識、後者は規程の知識であり、どちらも申請の本題（何を買うか）とは別に必要です。この二つを埋めるだけなら、AI に代わりを頼めるのではないのでしょうか。

実際の購買申請業務を題材にしますが、本チュートリアルで作るアプリケーションは、この既存画面を改修するのではなく、そこから独立した参考実装です。

本チュートリアルでは、「[まず動かす（クイックスタート）](#)」で作った最小のエージェントを、この見立てのとおり育てます。「購買申請支援」という業務課題を題材に、「[本格構築を始める前に](#)」以降の各章で個別に扱う機能——ナレッジベース、ツール、構造化出力、ガードレール、精度評価、業務への組み込み——を、判断の理由を添えながら一度ずつ使います。各節では、機能を選ぶ理由と設定の考え方を示し、選択肢が分かれる箇所はそれぞれの機能を扱う章の該当節へ誘導します。

題材の切り出し方

この業務課題は、見積書の読み取りと勘定科目の推定、理由文の生成、承認見込みの判定という3つの段階からなります。このうち本チュートリアルで作るのは、最初の2段階に絞ります。起点は見積書のテキストとします。単体のアプリケーションが見積書ファイルから抽出したテキストを、エージェントの入力としてそのまま渡す設計にし、画像だけの見積書からのテキスト抽出は本チュートリアルの対象外とします（「[発展](#)」で扱います）。

過去の類似申請を条件検索するには専用のロジックツールが必要ですが、本チュートリアルでは作りません（発展で扱います）。理由文の根拠は過去の申請ではなく、購買承認規定のナレッジに求めます。承認ルートの最終決定は既存の購買申請システムに任せ、エージェントは購買承認規定と照合した該当条項を示すだけにとどめ、申請を送信しません。この線引きは、「[業務を設計する](#)」で決めます。

項目

- [このチュートリアルで作るもの](#)
- [業務を設計する](#)
- [ナレッジベースを用意する](#)
- [ツールを用意する](#)
- [エージェントを定義する](#)
- [テスト実行で確かめる](#)
- [ガードレールを足す](#)
- [精度評価する](#)
- [公開して業務に組み込む](#)
- [運用を始める前に](#)
- [発展](#)

このチュートリアルで作るもの

本チュートリアルで作るのは、購買申請支援を行う単体のアプリケーションです。IM-Workflow のワークフローとは直接の関係を持たないため、IM-Workflow を導入していなくても動作は確認できます。見積書をアップロードして実行すると、勘定科目の候補、3種類の理由文、購買承認規定との照合結果と注意点を返します。申請者はこれを申請フォームに転記します。判断そのものを渡すのではなく、判断の材料を揃えるところまでを任せる、という線引きです。

前提として、「[まず動かす（クイックスタート）](#)」を完了していること、ベクトルデータベース接続設定とジョブネットの有効化が済んでいることが必要です。配布し

たエージェント定義を取り込む場合は、AIモデル管理に ID `gpt-5.4` のテキスト生成モデルも登録してください。クイックスタートで別の ID を使った場合は、ここで追加登録できます。配布する資料は次のとおりです。使い方は、それぞれ資料を使う節にあります。

資料	使う節
account_master.xlsx	「 ナレッジベースを用意する 」
purchase_regulation.md	「 ナレッジベースを用意する 」
vendor_master.json	「 ツールを用意する 」
インプットパラメータの一括編集用 JSON (agent_input_schema.json)	「 エージェントを定義する 」
アウトプットパラメータの一括編集用 JSON (agent_output_schema.json)	「 エージェントを定義する 」
quotation_sample.pdf	「 テスト実行で確かめる 」
評価データセット (json)	「 精度評価する 」
エージェント定義のエクスポート (json)	「 発展 」
参考実装一式 (zip)	「 公開して業務に組み込む 」

このほか、動作確認済みのナレッジベースのエクスポート ([knowledge_data.zip](#)) とロジックツールのエクスポート ([logic_tool_data.json](#)) も、自分の登録内容と見比べるための参考資料として配布しています（ツールのエクスポートには参照先のロジックフロー定義が含まれないため、単体では動作しません）。

業務を設計する

エージェントを定義する前に、「[エージェント定義の画面を開く前に決めること](#)」で挙げる 5 つの項目を、この業務課題に当てはめて決めておきます。

- 誰の何を助けるか: 申請者が申請書を「書く」作業を助けます。承認するかどうかの判断は助けません。
- 入力と出力: 入力は見積書のテキスト一つ、出力は構造化された提案です。
- ナレッジかツールか: 勘定科目マスタと購買承認規定は、全社共通で頻繁には変わらない規程であり、ナレッジとして持たせます。取引先マスタは日々更新される業務データであり、実行のたびに最新の状態を問い合わせる必要があるため、ツールとして持たせます。
- どこから呼ぶか: ロジックフロータスクとして呼びます。
- 期待する動き: 見積書のテキストから品目・数量・単価・取引先名を正しく読み取り、勘定科目候補を提案する。取引先マスタ検索ツールで取引先の登録状況を確認し、購入先選定理由に反映する。購買承認規定と照合して該当条項を示すが、承認ルートの可否は確定しない。この 3 つの期待は、そのまま「[精度評価する](#)」で精度評価の期待値として使います。

この段階で、AI は提案するだけで申請を送信しない、という方針を決めておきます。承認ルートの判定を含めてエージェントに委ねてしまうと、誤った提案がそのまま業務に影響する範囲が広がるためです。

ナレッジベースを用意する

「勘定科目マスタ」（ナレッジベース ID `8i5k3a4i8srt81i`）と「購買承認規定」（ナレッジベース ID `8i5k3b53osrtm1i`）の、2 つのナレッジベースを作ります。作成画面では自動入力された ID をそれぞれこの値に書き換えてから保存します。性質の異なる情報を一つにまとめない理由は「[粒度を決める](#)」で扱います。

サンプルファイルを手動アップロードし、「説明」（人が読む、多言語対応の項目）と「AIエージェント向けの説明」（LLM が読む、多言語非対応の項目）を書き分けます。「AIエージェント向けの説明」には、いつこのナレッジベースを使うべきかを書きます。たとえば購買承認規定であれば、「申請金額と承認ルートの対応を調べるときに使う」のように書きます。

チャンク分割の設定も、既定値（最大チャンク長 1024 文字、オーバーラップ 200 文字）から変更します（基準は「[チャンク分割を設定する](#)」を参照）。既定値のままでは、このサンプルの分量では 1 チャンクに収まってしまい、チャンク分割結果で複数チャンクに分かれる様子を確認できません。

- 勘定科目マスタ: 科目ごとの説明が短く独立しているため、最大チャンク長 150 文字・オーバーラップ 20 文字にする
- 購買承認規定: 金額帯ごとの例外条件をひとまとまりで読ませたいため、最大チャンク長 400 文字・オーバーラップ 50 文字にする

ベクトル化はジョブで実行されるため、登録した直後は検索できません。ジョブネットが有効になっていること、そしてアイテムの状態が「完了」になっていることを確認してから次に進みます。ここでうまくいかない場合の切り分けは「[ジョブネットの作成と有効化](#)」を参照してください。

ツールを用意する

本チュートリアルでは、「取引先マスタ検索」ロジックツール（「[ロジックツールを作る](#)」）を、ツール ID `prassist_vendor_lookup_flow` で 1 つ作ります。取引先マスタはナレッジのように事前に取り込んで検索させる対象ではなく、実行のたびに最新の登録状況を問い合わせる必要があるため、ツールとして持たせます。

処理本体のロジックフロー `prassist_vendor_lookup_flow` は、「[このチュートリアルで作るもの](#)」の参考実装一式に含まれています。参考実装を導入してサンプルデータセットアップを行うと、このフローが取り込まれます。

参照するロジックフローには `prassist_vendor_lookup_flow` を指定します。フローの入力パラメータ `vendorName` に取引先名を 1 つだけ渡します。見積書のテキストから取引先名を読み取るのは LLM の役目とし、ツール自身は受け取った取引先名で照会し、登録の有無と登録済みなら管理コードを返すだけにします。プロパティ（人が事前に決める値）と入力パラメータ（LLM が実行時に埋める値）の違いは、この 1 パラメータの設定で確認できます。



コラム

取引先マスタ検索ツールは参照のみで、登録・更新は行いません。副作用のあるツールを扱う場合の考え方は「[副作用のあるツールの扱い方](#)」で扱います。

エージェントを定義する

「[最小のエージェントを作る](#)」で作ったエージェントに、設定を積み増していきます。

まず「入出力値」（「[入出力値を設計する](#)」）を開き、次のように設定します。項目を個別に追加するほか、「一括編集」で配布した JSON の内容を貼り付けることもできます。

1. `input` の「一括編集」を開き、[agent_input_schema.json](#) の内容をすべてコピーして、編集欄の JSON と置き換え、「適用」をクリックします。個別に追加する場合は、インプットパラメータ `quoteText` を型 `string`、説明「見積書から抽出したテキスト」、必須で設定します。
2. 「構造化出力」をオンにします。`response` の「一括編集」を開き、[agent_output_schema.json](#) の内容をすべてコピーして、編集欄の JSON と置き換え、「適用」をクリックします。個別に追加する場合は、次の表の出力パラメータを設定します。いずれも必須です。

購買申請支援エージェントの出力パラメータ

パラメータ名	型	説明・子フィールド
<code>vendorName</code>	<code>string</code>	見積書から読み取った取引先名（購入先）。読み取れない場合は空文字にする
<code>totalAmount</code>	<code>number</code>	見積書に記載された税抜合計金額。読み取れない場合は 0 にする
<code>lineItems</code>	<code>array</code> （子フィールドを追加する）	見積書の明細行。品目が読み取れない場合は空配列にする。子フィールドは <code>itemName</code> （ <code>string</code> ）、 <code>quantity</code> （ <code>integer</code> ）、 <code>unitPrice</code> （ <code>number</code> ）、 <code>subtotal</code> （ <code>number</code> ）
<code>accountCandidates</code>	<code>array</code> （子フィールドを追加する）	勘定科目候補。子フィールドは <code>accountCode</code> 、 <code>accountName</code> 、 <code>itemName</code> 、 <code>reason</code> （いずれも <code>string</code> ）
<code>businessReason</code>	<code>string</code>	見積書から読み取れた実施理由・背景を書く。記載がなければ、その旨と申請者が記入すべき観点（新規導入・買い替え・増員など）を示す。承認可否や権限の注記は含めない
<code>accountReason</code>	<code>string</code>	勘定科目マスタとの対応関係に基づく物品選定理由のみを書く。承認可否や権限の注記は含めない
<code>vendorReason</code>	<code>string</code>	取引先照会ツールの結果に基づく購入先選定理由のみを書く。承認可否や権限の注記は含めない
<code>policyStatus</code>	<code>string</code>	購買承認規定との照合状態。 <code>MATCH</code> （該当条項が明確に一致）、 <code>REVIEW</code> （要確認）、 <code>UNKNOWN</code> （判定不能）のいずれか。承認結果を意味しない
<code>warnings</code>	<code>array</code> （子フィールドなし）	説明は未設定
<code>evidence</code>	<code>array</code> （子フィールドなし）	説明は未設定

`lineItems` と `accountCandidates` の子フィールドもすべて必須にします。子フィールドのない `array` は、文字列の配列として扱われます。

表の説明は要約です。説明文まで配布定義に合わせる場合は、一括編集用 JSON の内容を使用してください。

次に「指示」（システムプロンプト）に、次の文章をそのまま入力します。`${quoteText}` はインプットパラメータ `quoteText` の値を実行時に埋め込むテンプレート変数です（「[指示を書く](#)」）。

あなたは購買申請の記入支援者です。入力 `${quoteText}` は見積書から抽出したテキストです。ここから取引先名、品目、数量、単価、合計を読み取ってください。取引先照会ツールで、読み取った取引先名の登録状況を確認してください。勘定科目マスタから品目ごとの候補を探し、購買承認規定から該当条項を探してください。実施理由・背景、物品選定理由、購入先選定理由を別々に書いてください。ツールまたは検索に失敗した部分は飛ばして残りを処理し、欠けた事実を推測で補わないでください。承認可否の判定・申請・承認・発注・データ更新は行わないでください。この制約はあなたの行動についてのものであり、理由文にその旨を書く必要はありません。``response`` のスキーマで返してください。

指示を入力したら、次の順に割り当てと実行制御を設定します。

1. 「ナレッジベース」で、先ほど作成した「勘定科目マスタ」と「購買承認規定」を選びます。どちらも Top-K と類似度閾値は未設定のままにします（「[ナレッジベース](#)」）。
2. 「ツール」で、前節の「取引先マスタ検索」ロジックツールを選び、有効にします（「[ツール](#)」）。
3. 「実行制御」で、ストラテジが「シンプル」のままであることを確認して保存します。ストラテジの選び方は「[実行制御を設定する](#)」を参照してください。

テスト実行で確かめる

ここで使う見積書は、「[このチュートリアルで作るもの](#)」で配布した [quotation_sample.pdf](#) です。編集画面右側のテスト実行欄には PDF を添付できないため、PDF から抽出したテキストを入力します。次の内容をコピーして使えます。

見積書

見積書番号: DEMO-Q-001

発行日: 2026-01-15

取引先: 湊テクノロジー株式会社

宛先: 青柳 辰巳 様

下記のとおりお見積り申し上げます。

品目	数量	単価 (円)	小計 (円)
ノートPC	2	180,000	360,000
ドッキングステーション	2	20,000	40,000
税抜合計 (円) : 400,000			

- エージェントの編集画面を開いたまま、右側の「テスト実行」パネル下部の入力欄に、上の見積書テキストを貼り付けます。送信ボタンをクリックして、応答が完了するまで待ちます（「[テスト実行で確かめる](#)」）。
- 応答の構造化出力を確認します。`vendorName` が「湊テクノロジー株式会社」、`totalAmount` が 400000 で、`lineItems` にノートPCとドッキングステーションの2行が含まれることを確かめます。`accountCandidates`、3種類の理由文、`policyStatus`、`warnings`、`evidence` も出力されているか確認します。生成結果の文面や候補は毎回同じになるとは限りません。
- 応答の下に表示されるトレースID をクリックします。トレースの参照権限があれば、今回の実行の詳細画面が別タブで開きます。記録は非同期のため、直後に「見つかりません」と出た場合は少し待ってから開き直します。
- トレース詳細のスパンツリーとウォーターフォールで、モデル呼び出し、取引先マスタ検索ツールの呼び出し、ナレッジ検索をたどります。ツール呼び出しの詳細では、入力取引先名と、登録状況・管理コード（サンプルでは `registered`・`DEMO-V-001`）を確認します。ナレッジ検索の詳細では、勘定科目マスタと購買承認規定から検索結果が返っているか確認します。画面の読み方は「[トレース一覧・詳細の読み方](#)」を参照してください。

コラム

編集画面のテスト実行欄は、入力テキストをユーザメッセージとして送ります。`inputParameters.quoteText` は渡さないため、このテストでは指示中の `${quoteText}` が実際の値へ置き換わることまでは確認できません。インプットパラメータの受け渡しは、公開後に「[公開して業務に組み込む](#)」のロジックフローから実行して確認します。

つまづいたときは、症状から切り分けます。

- 取引先マスタ検索を呼ばずに購入先選定理由を返す場合は、トレースにツール呼び出しのスパンがあるかを確認します。無ければ、「[エージェントを定義する](#)」の指示を「見積書から取引先名を読み取れたら、購入先選定理由を書く前に『取引先マスタ検索』ツールを呼び、`vendorName` にその取引先名を渡してください。ツールの結果を受け取ってから理由文を書いてください」のように具体化し、再度テスト実行します。
- 取引先マスタ検索ツールは呼ばれているのに結果が違う場合は、トレースのツール呼び出しの詳細で `vendorName` が見積書の取引先名と一致するか確認します。一致しなければ指示とツールの入力パラメータ定義を見直し、一致していれば取引先マスタのデータとロジックフローを確認します（「[ツールを用意する](#)」）。
- ナレッジ検索がゼロヒットになる場合は、ナレッジベースの準備（「[ナレッジベースを用意する](#)」）に戻ります。

ガードレールを足す

見積書のテキストにメールアドレスや電話番号の書式が含まれていたら「記録する」ガードレールを追加します。配布したエージェント定義と同じ設定です。入力の検査位置では利用者のメッセージ本文を検査します（「[検査位置を選ぶ](#)」）。

- エージェントの編集画面で「ガードレール」を開き、「ガードレールを追加」をクリックします。
- 名称に「見積書の個人情報検知」、説明に「見積書のテキストにメールアドレスや電話番号の書式が含まれていないか確認します。」を入力します。検査位置は「入力」を選び、ガードレールを有効にします。
- 「判定スクリプト」に次のコードをそのまま貼り付けます。

```
function run(input, ctx) {
  var emailPattern = /[a-zA-Z0-9_+.-]+@[a-zA-Z0-9-]+\.[a-zA-Z0-9-]+/;
  var phonePattern = /0\d{1,4}-\d{1,4}-\d{3,4}/;
  if (emailPattern.test(input) || phonePattern.test(input)) {
    return { action: "LOG", message: "見積書のテキストに個人情報らしき記載を検知しました" };
  }
  return { action: "PASS" };
}
```

- 保存前に「テスト」を開き、「検査文字列」に `見積書担当: demo@example.invalid` を貼り付けて「テスト実行」をクリックします。結果が「記録する」で、文言が「見積書のテキストに個人情報らしき記載を検知しました」になることを確認します。
- 検査文字列を `見積書番号: DEMO-Q-001` に置き換えて再度「テスト実行」をクリックします。今度は「何もしない」になることを確認します。テスト機能ではエージェント本体は実行されません（「[保存前にテストする](#)」）。
- テスト画面を閉じ、ガードレールの設定を確定してから、エージェントを保存します。

この例の正規表現は書式だけを判定するため、個人情報であるかどうかを厳密に判定するものではありません。まず「記録する（LOG）」で発動状況を確認し、誤検知が許容できる場合に限り、スクリプト内の `action: "LOG"` を `action: "BLOCK"` に変更して再テストします。「止める（BLOCK）」にするとエージェントの実行が中止されます（「[動作を選ぶ](#)」）。配布したエージェント定義は LOG のままです。

「記録する」の判定結果は監査ログには残りません。発動はトレースで確認します。

精度評価する

作業手順評価（「[評価種別の選び方](#)」）の計画を作り、「[このチュートリアルで作るもの](#)」で配布した評価データセット（10 件。期待するツール・ナレッジベースの呼び出しと期待回答が含まれる）を取り込んで実行します。期待呼び出しは表示名ではなく ID で照合されるため、上記のナレッジベースID とツールID、入力パラメータ名 `vendorName` を確認してください。このデータセットには、「[業務を設計する](#)」で書き出した 3 つの期待が、そのままテストケースとして入っています。

指標はツール正確性、タスク完遂率、最終出力の品質（読み方は「[エージェントの指標から調べる箇所](#)」）で読みます。評価結果に添えられる改善提案の使い方もここで確認します。実行制御を「Plan & Action」に切り替えて再実行し、指標がどう変わるかを比べておくと、ストラテジの違いが数値で見えてきます。

公開して業務に組み込む

まず Accel Agent の編集画面で「基本情報」の用途「ロジックフロータスク」を ON にし、画面右上の「保存」、続いて「公開」をクリックします。公開済みのエージェントだけをロジックフローから実行できます（「[公開して IM-LogicDesigner から呼ぶ](#)」）。

ここからは IM-LogicDesignerの画面で作業します。以下の手順は、参考実装一式（[zip](#)）に含まれる `prassist_purchase_flow`（「購買申請支援」）を手作業で作る場合の設定です。参考実装を導入してサンプルデータセットアップを行った場合は同じフローが取り込まれるため、設定を見比べながら進められます。

- フロー定義を新規作成し、フローID を `prassist_purchase_flow`、フロー名を「購買申請支援」にします。参考実装を取り込んでいる場合は、同じ ID のフローを重複して作らず、取り込まれた定義を開きます。
- フローの入力データ定義に、必須の文字列 `quoteText` を追加します。ここへ PDF から抽出した見積書のテキストを渡します。フローの出力データ定義には、「[エージェントを定義する](#)」で設定したエージェントの構造化出力と同名・同型の `vendorName`、`totalAmount`、`lineItems`、`accountCandidates`、`businessReason`、`accountReason`、`vendorReason`、`policyStatus`、`warnings`、`evidence` を追加します。配列とその子項目も、同節の出力パラメータ表に合わせます。
- フローの定数に、文字列 `PROMPT_TEXT` を追加し、値を「見積書の内容を確認して支援結果を作成してください。」にします。これはエージェントを起動するためのメッセージです。見積書の本文はこの定数に含めません。
- パレットの「IM-Copilot」カテゴリから「エージェント実行」タスクを配置し、開始→エージェント実行→終了の順につなぎます。タスクの「タスク固有設定」で、エージェントID の検索アイコンからこのチュートリアルで作ったエージェントを選び、「常に最新のバージョンを利用する」を選択します（「[IM-LogicDesigner の「エージェント実行」タスク](#)」）。
- エージェント実行タスクのマッピングで、フロー入力 `$input/quoteText` を `inputParameters/quoteText` に接続します。さらに `prompts` に要素を 1 件追加し、その `text` に定数 `$const/PROMPT_TEXT` を接続します。`prompts` を空にするとエラーが発生します。
- 終了タスクのマッピングで、エージェント実行タスクの `result/outputParameters` にある各項目を、フローの同名の出力項目へ接続します。例えば `result/outputParameters/vendorName` → `$output/vendorName`、`result/outputParameters/accountCandidates` → `$output/accountCandidates` です。手順 2 で追加した 10 項目すべてを対応付けます。`result/message` ではなく、構造化出力の `outputParameters` を使います。
- フローを保存してデバッグ実行します。入力データの `quoteText` に「[テスト実行で確かめる](#)」の見積書テキストを貼り付けます。「フロー定義の変数情報」の「出力」で `vendorName` が「湊テクノロジー株式会社」、`totalAmount` が 400000、`lineItems` に 2 行があることを確認します。生成される文面や勘定科目候補は毎回同じとは限りません。

注意

ロジックフローのデバッグ実行でも、エージェントの実行は「本番」として記録され、AI モデルの課金やツールの処理が実際に発生します。このチュートリアルの取引先マスタ検索は参照のみですが、他のツールに置き換えた場合は副作用にも注意してください。

このロジックフローを呼び出す画面自体の実装は本チュートリアルの範囲外です。見積書のアップロードから支援結果の確認までを行う「購買申請支援」という参考実装を、画面・ロジックフロー定義一式として「[このチュートリアルで作るもの](#)」で配布しているので、実装の参考にしてください。

図：参考実装の「購買申請支援」画面。見積書をアップロードして「AIによる支援を実行」すると、勘定科目と3種類の理由文の推定結果が表示される。

運用を始める前に

本番投入の前に、次の 4 点を確認します。ジョブネットが本番環境でも有効になっているか（「[ジョブの運用](#)」）、AIモデルの単価が入っているか（「[AIモデル管理](#)」）、運用ダッシュボードで実行が確認できるか（「[運用ダッシュボード](#)」）、監査ログに公開の記録が残っているか（「[監査ログ](#)」）です。

発展

ここまでで一通り動くものができました。時間があれば、次の発展も試してみてください。

- 既存の購買申請画面と統合する（本チュートリアルで作った支援機能は単体のアプリケーションでしたが、IM-Workflow の申請画面に見積書アップロードと支援結果の表示を組み込めば、申請者は画面を行き来せずに済みます。IM-Workflow の導入と画面開発は本チュートリアルの範囲外のため、各製品のガイドを参照してください）
- 別の環境へ移す（「[インポート／エクスポート](#)」。「[このチュートリアルで作るもの](#)」で配布したエージェント定義のエクスポートを取り込む手順を兼ねます。取り込み先にも、上記の ID の AIモデル・ナレッジベース・ロジックツールを先に用意します）
- 承認側を助ける（購買申請の承認判断を支援するエージェント。人の操作ではなく申請の提出という業務イベントをきっかけに、IM-Workflow のシステムノード（人の操作を介さず自動的に処理が進むノード）内のIM-LogicDesignerのフローから、同じ「エージェント実行」タスクで呼び出します）
- 見積書の画像を読ませる（テキスト抽出に失敗する画像だけの見積書は、抽出前の画像を `prompts` の `image` でエージェントに渡す設計に変えられます。動作保証モデルのうちマルチモーダルに対応するものを選びます）
- 過去の類似申請を参照する（条件検索の組み込みツールは無いので、取引先マスタ検索と同様にロジックツールを新たに作ります）
- 会話の続きを扱う（エージェントセッションの取得・保存・分岐タスクを使います）

本格構築を始める前に

「[チュートリアル: 購買申請支援エージェントを業務に組み込む](#)」で一通りの流れをつかんだら、次はナレッジベース・ツール・スキル・エージェント定義・ガードレールを、業務に合わせて本格的に作り込みます。この 5 つはそれぞれ別の章で扱います。まず本章で最初の判断を済ませ、その結果に応じて読む章を決めます。

エージェント定義の画面を開く前に決めること

エージェント定義の画面を開く前に、必須項目から逆算して次の 5 点を決めておきます。

- 1. 誰の何を助けるか。エージェントの指示に書く役割そのものです。
- 2. 入力と出力。インプットパラメータや構造化出力が必要かどうかは、ここで決まります。
- 3. 判断に使う情報を、ナレッジベースとツールのどちらで持たせるか。
- 4. どこから呼び出すか。ロジックフロータスクか、BPMタスクか、または両方か。
- 5. 期待する動き。この期待は、そのまま精度評価のテストケースに使えます。

決める内容は業務ごとに異なるため、本書では決めるべき項目だけを示します。記入例は「[業務を設計する](#)」を参照してください。

この 5 点のうち、項目 3 でナレッジベースかツールかを決めた結果が、「[ナレッジベース](#)」・「[ツール](#)」のどちらを先に読むかを決めます。項目 1 の役割と項目 4 の呼び出し元は、「[エージェント定義](#)」で指示・基本情報として書きます。

読む順序の目安

本格構築で扱う章は、材料を用意してから組み立てる、という順序です。

章	その章で決めること	読む場面
「 ナレッジベース 」	ナレッジベースの設計・整備	判断材料を検索で持たせる場合
「 ツール 」	ツールの選定・作成	判断材料や実行手段を組み込み・ロジック・MCPで持たせる場合
「 スキル 」	スキルの作成	手順書をLLMに読ませる場合
「 エージェント定義 」	エージェント定義の全セクション	材料が揃った後、エージェントを組み立てる場面
「 ガードレール 」	ガードレールの設定	エージェントの入出力を検査・制限したい場面

ナレッジベース・ツール・スキルのいずれも使わない、指示だけの単純なエージェントであれば、これらを読み飛ばして「[エージェント定義](#)」から始めて構いません。公開後の改善は、新しいバージョンとして保存してから精度評価（「[精度評価](#)」）を行う、という順序で回します。

ナレッジベース

ナレッジベースは、エージェントが意味検索で参照する情報源です。文書を取り込んでチャンクに分割し、ベクトル化して検索できる状態にします。

本章では、初めて整備する担当者向けに、設計の考え方から取り込み・分割・メタデータ・運用までを説明します。画面の場所と個々の入力項目も本章で説明しているため、本章を見ながら作業を進められます。

項目

- ナレッジベースで確かめること
 - 製品がやること・利用者がやること
 - 取り込めるファイル
 - 精度は検索と推論の掛け算で決まる
 - 画面を開く
- ナレッジベースを設計する
 - 粒度を決める
 - 2つの説明欄を書き分ける
- アイテムを取り込む
 - 詳細設定で指定できること
 - 登録後に確認すること
- チャンク分割を設定する
 - 再ベクトル化のやり方
- メタデータ項目を設計する
 - メタデータが使われる場面
 - ビルトイン項目とカスタム項目
- 有効期間で検索対象を管理する
- 埋め込みモデルの変更とナレッジベースの削除
- インデックス状況と利用箇所を確認する
- インポート・エクスポート
- 精度が出ないときの切り分け
- 困ったときの確認

ナレッジベースで確かめること

製品がやること・利用者がやること

製品が自動で行うのは、テキスト抽出、チャンク分割、ベクトル化の3工程だけです。文書の整形、OCR、Q&A形式への変換は、intra-mart Accel Platformの外で事前に済ませておく必要があります。

取り込めるファイル

手動アップロードで取り込めるファイルの形式と、形式ごとのサイズの上限は、テキスト抽出設定（ND Universal Extractorの設定ファイル`im-nd-extractor-config.xml`）で決まります。取り込む前に、利用する環境のテキスト抽出設定で対象の形式と上限を確認します。設定の内容は、「[設定ファイルリファレンス](#)」の「[テキスト抽出設定](#)」を参照してください。

テキスト抽出設定にない形式でも、MarkdownのようにMIMEタイプが`text/`で始まるファイルは、UTF-8のテキストとしてそのまま読み込みます。また、テキスト抽出設定の上限にかかわらず、100MBを超えるファイルは取り込めません。

アイテムを追加する画面では、ファイルを選ぶ時点では形式を絞り込みません。そのため、取り込めないファイルも選択できます。取り込めないファイルは、次のどちらかの段階でエラーが発生します。

- **アイテムの追加時:** 形式がテキスト抽出設定にない場合や、サイズが上限を超える場合は、アイテムを追加する操作がエラーメッセージを表示して失敗します。アイテムは登録されません。
- **ベクトル化の実行時:** 形式とサイズが条件を満たしていても、ファイルからテキストを取り出せない場合は、ベクトル化の段階で失敗し、アイテムの状態が「エラー」に変わります（「[登録後に確認すること](#)」）。文字を画像としてだけ持つPDF（スキャンした書類など）や、破損したファイルが該当します。

元資料は、登録する前にこれらの条件に収まるかを確認して用意します。スキャンした書類のように文字が画像になっている資料は、OCRでテキストを付けたPDFにするか、テキストに起こしてから登録します。

精度は検索と推論の掛け算で決まる

エージェントの回答精度は、ナレッジベースの検索精度と、エージェント側の推論品質の掛け合わせで決まります。検索が的確でも推論で読み違えることがあり、逆に推論が的確でも検索が取りこばせば答えがありません。精度が出ない場合は、まずナレッジの設計・前処理・カバレッジを疑い、その後で技術的なチューニングに進みます（「[精度が出ないときの切り分け](#)」）。

画面を開く

サイトマップから「Copilot」→「Accel Agent」→「ナレッジベース管理」を開きます。

一覧画面では、ナレッジベースごとの状態を「すべて／正常／要確認」で絞り込めます。「要確認」には、アイテム単位のエラーやメタデータ同期失敗（「[登録後に確認すること](#)」）を含むナレッジベースが集約されます。個々のアイテムまで見るには、対象のナレッジベースを開きます。

ナレッジベースを設計する

粒度を決める

ナレッジベースはカテゴリで整理したうえで、業務ドメインごとに分けます。一つにまとめすぎると、検索結果に無関係なチャンクが混ざります。逆に細かく分けすぎると、検索範囲が狭くなりカバレッジが不足します。

例えば、購買申請を支援するエージェントであれば、「勘定科目マスタ」と「購買承認ポリシー」を 1 つのナレッジベースにまとめません。前者は科目の対応表という事実の集まり、後者は承認ルートを説明する規程文であり、性質も更新頻度も異なるためです。まとめてしまうと、科目を尋ねただけの質問に承認ルートの文章が紛れ込み、逆に承認ルートを尋ねた質問に無関係な科目の羅列が混ざります。

ナレッジベース名（255 文字まで）と「説明」（1000 文字まで）は、この粒度の分け方が一覧で見分けられるように付けます。「説明」は「AIエージェント向けの説明」（「[2 つの説明欄を書き分ける](#)」）と違い、人が一覧で識別するための文章です。

2 つの説明欄を書き分ける

ナレッジベースには、人が読む「説明」と、LLM が読む「AIエージェント向けの説明」（入力欄の名前は「内容と用途」）の 2 つの説明欄があります。多言語で保持できるのは、名前と「説明」だけです。「AIエージェント向けの説明」は多言語に対応しません。

LLM は、ナレッジベースを選ぶ際に「AIエージェント向けの説明」だけを手がかりにします。対象を述べるだけでなく、いつ使うべきかを書く必要があります。

書き方	例
対象を述べるだけ（不十分）	購買に関する規程
いつ使うべきかまで書く（推奨）	申請金額と承認ルートの対応を調べるときに使う

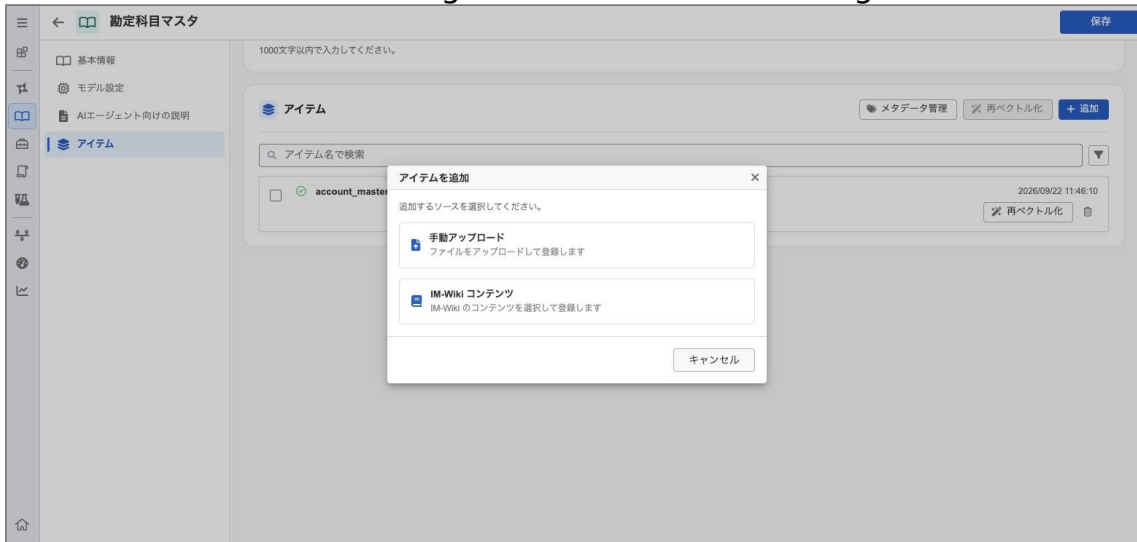
前者は「何についてのナレッジか」しか語っておらず、似た主題の別のナレッジベースがあると LLM はどちらを検索すべきか判断できません。後者は「どの場面で引くべきか」まで書いてあるため、複数のナレッジベースを割り当てたエージェントでも選び分けられます。

The screenshot shows the 'ナレッジベース編集画面' (Knowledge Base Edit Screen) for '勘定科目マスタ' (Account Subject Master). The left sidebar contains navigation links: '基本情報' (Basic Information), 'モデル設定' (Model Settings), 'AIエージェント向けの説明' (AI Agent Description), and 'アイテム' (Items). The main content area is divided into two sections: 'モデル設定' (Model Settings) and 'AIエージェント向けの説明' (AI Agent Description). The 'モデル設定' section includes a '埋め込みモデル' (Embedding Model) dropdown menu with 'text-embedding-3-small' selected. The 'AIエージェント向けの説明' section includes a '内容と用途' (Content and Usage) text area with a placeholder text: 'AIエージェントが、このナレッジベースを使うかどうかを判断する材料になります。対象となる業務や文書の種類を具体的に書いてください。未入力の場合は「ナレッジベース名」のナレッジベースという説明が使われます。' (AI agents use this knowledge base as material to decide whether to use it. Please specify the type of business or document it applies to. If not entered, the explanation 'ナレッジベース名' (Knowledge Base Name) will be used.) and a '見積書の品目と用途から勘定科目候補を提案するときだけ参照する。表にない品目は推測でコードを作らない。' (Only refer to it when proposing accounting subject candidates from the items and uses of the quotation. Do not create codes for items not in the table by inference.) at the bottom.

図：ナレッジベース編集画面。「説明」（人が読む）と「AIエージェント向けの説明」（LLM が読む）は別欄になっている。

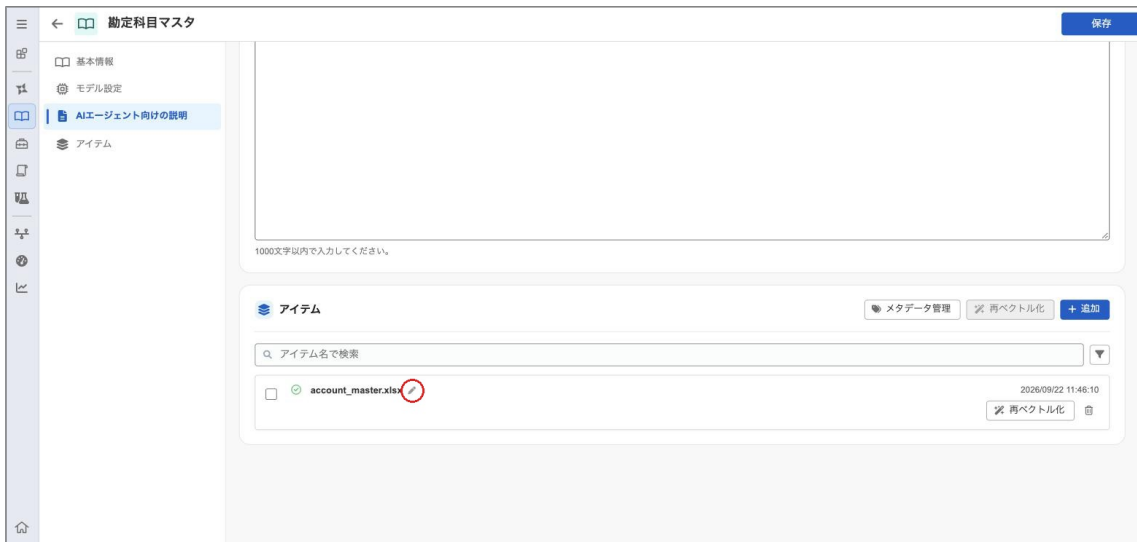
アイテムを取り込む

アイテムの追加は、ソースタイプ（手動アップロードか IM-Wiki コンテンツか）を選び、ファイルを選択する、または IM-Wiki コンテンツを選ぶだけで完了します。名前は選んだファイル名から自動的に入力されますが、編集できます。



図：アイテム追加画面。追加するソースタイプを選択する。

有効期間・外部ソース変更検知・チャンク分割は、この追加画面には出てきません。既定値のまま登録され、変更したい場合は、追加したアイテムを一覧の編集アイコンから開き、「アイテム設定」画面で変更します。



図：アイテム一覧。アイテム名の横にある編集アイコンから、「アイテム設定」画面を開く。

詳細設定で指定できること



図：「アイテム設定」画面。基本設定に有効期間開始日・終了日・外部ソース変更検知、チャンク分割に最大チャンク長・オーバーラップと、ベクトル化済みのチャンク分割結果が並ぶ。

設定項目	何を決めるか
------	--------

設定項目	何を決めるか
有効期間開始日・終了日	検索対象に含める期間を絞りたい場合に設定する（任意・既定は無期限）。開始日は「この日から検索させたい」、終了日は「この日までしか検索させたくない」という指定
外部ソース変更検知	IM-Wiki コンテンツでのみ設定できる。有効にすると、IM-Wiki 側の更新を検知して自動的に反映する（任意・既定は無効）
チャンク分割	最大チャンク長・オーバーラップを調整する（「 チャンク分割を設定する 」）

アイテムの内容を更新する方法は、ソースタイプによって異なります。IM-Wiki コンテンツで外部ソース変更検知を有効にしていない場合は、アイテム一覧の「再ベクトル化」を実行すると、その時点の IM-Wiki の内容で登録し直されます（「[再ベクトル化のやり方](#)」）。手動アップロードしたアイテムはファイルを差し替えられないため、アイテムを削除してから登録し直します。削除すると、そのアイテムに設定した有効期間やメタデータも消えるため、登録し直した後に設定し直します。

登録後に確認すること

登録した後は、次の 2 点を確認します。

- **ジョブネットが有効になっているか:** 通常、ベクトル化は「ナレッジベースベクトル化」のジョブが行うため、登録した直後はまだ検索できません。ジョブネットの作成・有効化は環境構築時の作業で、済んでいなければベクトル化がいつまでも進みません（「[ジョブの運用](#)」）。ジョブを待たずにすぐ検索できるようにしたい場合は、アイテム一覧の「再ベクトル化」ボタンでその場で処理できます（「[再ベクトル化のやり方](#)」）。外部ソース変更検知や有効期間を設定したアイテムでは、それぞれに対応するジョブ（「ナレッジアイテム外部ソース変更検知」「ナレッジアイテム有効期間チェック」）も有効にします。これらのジョブが動いていないと、IM-Wiki 側の更新や有効期間の開始日を迎えてもアイテムが「待機中」にならないため、ベクトル化のジョブが動いていても反映されません。また、有効期間の終了日を過ぎても、アイテムは検索対象から外れません。
- **アイテムの状態が「完了」になっているか:** 一覧または詳細で状態を確認します。「エラー」になっている場合は、対応形式や内容を見直したうえで、アイテム一覧の「再ベクトル化」ボタンで再試行します。

アイテムの状態は、次のように表示されます。

状態	意味
待機中	ベクトル化ジョブの実行を待っている
処理中	ベクトル化ジョブが実行中
完了	ベクトル化が完了し、検索対象になっている
エラー	テキスト抽出・分割・ベクトル化のいずれかで失敗した
無効	有効期間終了日を過ぎ、検索対象から外れている
開始待ち	有効期間開始日が未来のため、まだ検索対象ではない
メタデータ同期失敗	メタデータの反映に失敗した

「無効」と「開始待ち」は、どちらもエラーではなく有効期間による正常な状態です。一覧で見分けがつきにくい場合は、対象アイテムの有効期間開始日・終了日を確認します。

チャンク分割を設定する

取り込んだ文書は、検索単位（チャンク）に分割してからベクトル化します。「[詳細設定で指定できること](#)」のとおり、この設定は「アイテム設定」画面にあります。当リリースで選べる分割方式は「再帰的分割」の 1 方式のみで、調整できるのは次の 2 項目です。どちらも文字数で指定します。

- **最大チャンク長:** 1 つのチャンクに含める文字数の上限です。文書は、空行（段落の区切り）、改行、空白の順に区切りを探しながら、この長さを超えないように分割されます。区切りが見つからない場合は、上限の位置で切られます。句点（。）は区切りとして扱われないため、改行の少ない文書では文の途中で分かれることがあります。
- **オーバーラップ:** 隣り合うチャンクに重ねて含める文字数です。チャンクの境目にまたがる内容が両方のチャンクに入るため、検索で取りこぼしにくくなります。その分、チャンクの数が増えます。

ベクトル化が完了すると、「アイテム設定」画面の「チャンク分割結果」で、実際に分割されたチャンクとそれぞれの文字数を確認できます。設定を変更した場合は、再ベクトル化が終わってから結果を確認します。



図：「アイテム設定」画面。ベクトル化済みのチャンクを、チャンク分割結果で確認できる。

分割の長さは、概念の説明が中心のナレッジなら大きめに、事実検索が中心のナレッジなら小さめに、という基準で決めます。例えば、購買承認ポリシーのように「例外規定とその条件」をひとまとまりで読ませたい文書は大きめのチャンク長に、勘定科目マスタのように「科目名と説明」が短い単位で独立している文書は小さめのチャンク長に向きます。

チャンク分割の設定を変更すると、保存した時点でそのアイテムはベクトル化待ち状態に更新されます（すぐに再ベクトル化されるわけではなく、ジョブの実行、または「再ベクトル化」ボタン（[「再ベクトル化のやり方」](#)）を待つ状態です）。なお、メタデータだけを変更した場合はこの再ベクトル化は発生せず、保存すると即時にベクトルデータへ付随するメタデータのみが更新されます。

再ベクトル化のやり方

アイテムが「待機中」になるケースは複数あります。

- アイテムを登録したとき
- チャンク分割設定を変更して保存したとき（変更前の状態によって異なります。後述の表を参照）
- 外部ソース変更検知で、IM-Wiki 側の更新が検知されたとき
- 有効期間の開始日を迎えたとき
- 「強制的に再ベクトル化待ちへ」ボタンをクリックしたとき

どの経路で「待機中」になった場合でも、実際にベクトル化されるのは次のどちらかのときです。

タイミング	場所	補足
ナレッジベースベクトル化のジョブの実行時	—	ジョブの実行周期はトリガの設定による（「 ジョブの運用 」）
「再ベクトル化」ボタンをクリックしたとき	アイテム一覧（複数選択して一括実行も可）	ジョブを待たず、その場で処理してすぐに結果が返る

すぐに検索できる状態にしたい場合や、「エラー」のアイテムを直した後にまず試す場合は、アイテム一覧の「再ベクトル化」ボタンを使います。「待機中」「完了」「エラー」「メタデータ同期失敗」のアイテムに使い、「処理中」と「無効／開始待ち」（有効期間外）のアイテムでは使えません。

「強制的に再ベクトル化待ちへ」は、「処理中」のアイテムを「アイテム設定」画面で開くと表示される、回復用のボタンです。システムダウンなどでベクトル化が中断されると、アイテムが「処理中」のまま残ることがあります。システム側では、実際に処理中なのか、中断されたまま残っているのかを判別できないため、このボタンは「処理中」のアイテムには常に表示されます。

このボタンは、アイテムの状態を「処理中」から「待機中」に書き換えるだけで、実行中のベクトル化を止めるものではありません。実際に処理中のアイテムに使うと、その処理が終わってもアイテムの状態は「完了」「エラー」に更新されず、「待機中」のまま残ります（処理が成功していれば、ベクトルデータ自体は作成されます）。そのため、クリックする前に、ナレッジベースベクトル化のジョブが実行中でないこと（ジョブネットモニタで確認します）と、ほかの利用者がそのアイテムの「再ベクトル化」を実行していないことを確認してください。クリックしても「待機中」に戻るだけなので、実際にベクトル化するには、ジョブの実行を待つか、「再ベクトル化」ボタンを使います。

チャンク分割設定を変更した場合、自動的に「待機中」へ戻るのは、変更前の状態が「完了」または「メタデータ同期失敗」だったアイテムだけです。

変更前の状態	チャンク分割設定を変更したときの挙動
完了／メタデータ同期失敗	自動的に「待機中」へ戻る
エラー	自動では戻らない。原因を解消したうえで、アイテム一覧の「再ベクトル化」ボタンで再試行する
無効／開始待ち	有効期間外のため、チャンク分割設定を変更しても「待機中」へは戻らない。有効期間内に戻った時点で「待機中」へ遷移する

「エラー」状態のアイテムがチャンク分割設定の変更だけでは復旧しないのは、原因を直さないまま自動的に再試行を繰り返さないようにするためです。エラーが続くアイテムに気づかず放置しないよう、インデックス状況（「[インデックス状況と利用箇所を確認する](#)」）でエラー件数を定期的に確認します。

メタデータ項目を設計する

Accel Agent 利用ガイド — Accel Agent 利用ガイド 初版 2026-10-01

メタデータ項目を定義する目的は、値をただ保持するだけでなく、検索やフィルタで実際に扱える形にすることにあります。例えば「登録日」という値を持たせても、それが日付型で範囲検索できるのか、文字列型で完全一致しか使えないのかを別途決めておかなければ、検索やエージェント実行時の絞り込みに使えません。値そのものとは別に、値の意味・型・使える演算子を定義しておく仕組みが、メタデータ項目定義です。

メタデータが使われる場面

メタデータは、次のような場面で使うことを想定して設計されています。

場面	使うメタデータ項目の例	できること
検索時のフィルタリング	業務種別、部門、文書種別、作成日	「購買」「自部門」「現行ルール」「直近1年」のように絞り込み、質問内容に合った文書を優先し、不適切な候補が混ざりにくくする
回答の根拠確認	元のファイル名、参照URL	生成した回答が、どのアイテムを根拠にしているかを利用者が確認できるようにする
Agentic RAG の再問合せ	対象部門、関連キーワード、業務種別、文書種別	1 回目の検索で得たメタデータを手がかりに、エージェントが検索条件を補って再検索する

このうち検索時のフィルタリングは、エージェント定義のメタデータフィルタ（「[ナレッジベース・スキル・ツール・ガードレールを割り当てる](#)」）として実際に設定できます。回答の根拠確認・Agentic RAG の再問合せは、メタデータを持たせておくこと自体が前提になる場面です。どのメタデータ項目を用意するかは、エージェントに何を確認させたいか・何を根拠として示したいかから逆算して決めます。

例えば、購買承認ポリシーのナレッジベースに、本社向けと支社向けで内容が一部異なる規程を混在させて登録する場合を考えます。`targetLocation`（対象拠点。選択肢型、候補は「本社」「支社A」「支社B」）というカスタム項目をアイテムごとに設定しておけば、エージェント側のメタデータフィルタと組み合わせて検索範囲を絞り込めます。フィルタの値は固定値のほか、エージェントのインプットパラメータへの参照にもできるため、「`targetLocation`=インプットパラメータの拠点コード」のように 1 つのエージェント定義のまま、呼び出し元が渡す拠点コードに応じて絞り込み先を変える、という組み方もできます。同じナレッジベースを、エージェントごとに、さらには呼び出しごとに違う範囲で使い分けられるのが、メタデータ項目を定義する利点です。

ビルトイン項目とカスタム項目

メタデータには、製品があらかじめ用意するビルトイン項目と、利用者が定義するカスタム項目があります。

ビルトイン項目	自動設定・読み取り専用
作成日時	原本の作成日時。IM-Wiki コンテンツでは IM-Wiki での作成日時、手動アップロードではファイルをアップロードした日時
最終更新日時	原本の最終更新日時。IM-Wiki コンテンツでは IM-Wiki での最終更新日時、手動アップロードではファイルをアップロードした日時
文書バージョン	将来の拡張に備えて用意している項目です。現在のソースタイプ（手動アップロード・IM-Wiki コンテンツ）では値は設定されません
元のファイル名	取り込んだファイルの名前
格納パス	取り込み元での格納場所（手動アップロードでは埋まらない）
参照URL	IM-Wiki コンテンツの参照先 URL（手動アップロードでは埋まらない）

格納パス・参照URLは、IM-Wiki コンテンツを取り込んだ場合にだけ自動で埋まります。手動アップロードでこれらの情報を絞り込みに使いたい場合は、カスタム項目として自分で定義する必要があります。

カスタム項目の型には、文字列・数値・日付・選択肢があります。型によって、エージェント側で設定できる絞り込み条件（演算子）が異なります。



図：アイテム一覧の「メタデータ管理」から開く「メタデータ管理」画面。選択肢型のカスタム項目を作成する例。項目名・型・説明・選択肢を指定する。

型	使える絞り込み条件	向いている項目の例
文字列	等しい（完全一致のみ。部分一致はできません）	ID・コードなど、表記が完全に定まる値
数値	等しい・等しくない・以上・以下・より大きい・より小さい・範囲内	金額・件数など大小を比較したい値
日付	等しい・以上・以下・より大きい・より小さい・範囲内	適用日・改定日など期間で絞り込みたい値
選択肢	等しい・等しくない・いずれかに一致	部署名や書類区分のように、表記ゆれを避けたい値

文字列型は完全一致でしか絞り込めないため、“本社”と“本社 ”（末尾に空白）のような入力ゆれがあると、フィルタで正しく拾えません。値の候補が事前に決まっているなら、選択肢型にしておくほうが運用で崩れにくくなります。項目名も型も登録後には変更できないため、最初の設計がそのまま運用の制約です。

制限	上限
1 ナレッジベースあたりの項目定義数	30
項目名の文字数	50 文字（英数字とアンダースコアのみ）
項目の説明文の文字数	500 文字
選択肢（オプション）1 件の文字数	30 文字
1 項目あたりの選択肢の数	100

有効期間で検索対象を管理する

有効期間の終了日を過ぎると、そのアイテムのベクトルデータは削除され、状態は「無効」に変わります。有効期間を変更して期間内に戻すと「待機中」に戻り、次のベクトル化で登録し直されます（「[再ベクトル化のやり方](#)」）。有効期間を設定しない場合、アイテムは期限なく検索対象です。

有効期間は、改定日が決まっている規程のように、検索させる期間があらかじめ決まっている文書に向いています。期間とは関係なく、アイテムを一時的に検索対象から外したり戻したりしたい場合は、メタデータで切り替えます。例えば、選択肢型のカスタム項目 `publishStatus`（公開状態。候補は「公開」「非公開」）を定義し、エージェントのナレッジベースの割り当てで「`publishStatus=公開`」のメタデータフィルタを設定します。メタデータだけの変更は再ベクトル化を伴わず、保存するとすぐに反映されるため、アイテムの `publishStatus` の値を変えるだけで検索対象を切り替えられます。この方法では、`publishStatus` を設定していないアイテムは検索されなくなるため、すべてのアイテムに値を設定します。また、そのナレッジベースを使うすべてのエージェントに、同じフィルタを設定しておく必要があります。

埋め込みモデルの変更とナレッジベースの削除

埋め込みモデルは、ナレッジベース単位で 1 つ設定します。変更すると、そのナレッジベースに属するアイテムのベクトルデータは削除され、全アイテムを再ベクトル化する必要があります。「完了」と「メタデータ同期失敗」のアイテムは自動的に「待機中」に戻りますが、再ベクトル化が終わるまでは検索できません。「エラー」のアイテムは自動では戻らないため、原因を解消してから「再ベクトル化」ボタンで再実行します。また、「処理中」のアイテムがある間には変更できません。異なる埋め込みモデルで作ったベクトルは互いに比較できないため、「一部だけ新しいモデルに切り替える」という段階的な移行はできません。このため、埋め込みモデルは最初の選定が重要です。

ナレッジベースは、エージェントから参照されている最中でも削除できます。削除する前に、利用箇所（「[インデックス状況と利用箇所を確認する](#)」）を確認する運用にしておく必要があります。

インデックス状況と利用箇所を確認する

インデックス状況は、ナレッジベース管理の一覧でナレッジベースを選ぶと表示される詳細の、「概要」タブで確認します。総アイテム数・完了率・エラー件数のほか、最近更新されたアイテムと、「エラー」または「メタデータ同期失敗」になっている要確認のアイテムが一覧で表示されるため、対応が必要なアイテムをすぐに見つけられます（状態の意味は「[登録後に確認すること](#)」を参照）。

利用箇所は、同じ詳細の「利用箇所」タブで確認します。そのナレッジベースを割り当てているエージェントと、利用しているバージョンが一覧で表示されます。

インポート・エクスポート

ナレッジベースは、配下のアイテムを含めてまとめてエクスポート・インポートできます。手順と注意事項は「[インポート／エクスポート](#)」を参照してください。配下のアイテムに、待機中・処理中・エラー・メタデータ同期失敗のいずれかが 1 件でも含まれる場合はエクスポートできません（「[エクスポートの注意点](#)」）。「無効」「開始待ち」は有効期間による正常な状態のため、これらが含まれていてもエクスポートは妨げられません。

精度が出ないときの切り分け

検索そのものを単体で試す画面は用意されていません。チャンク分割結果の確認と、トレースのナレッジ検索スパンを見ることで切り分けます（「[実行を追う（トレース）](#)」）。検索結果に必要な情報が含まれていなければナレッジ側の問題、含まれているのに回答が誤っていればエージェント側の推論の問題です。この切り分けは精度評価の指標（「[RAG の指標から調べる箇所](#)」）からも確認できます。

例えば、期限を尋ねた質問への回答が誤っていた場合、トレースのナレッジ検索スパンを開き、取得したチャンクに期限の記載が含まれているかを見ます。含まれていなければ、チャンク分割が期限の記述を別のチャンクへ分断していないか、「AI エージェント向けの説明」が的外れで別のナレッジベースが検索されていないかを疑います。含まれているのに回答が誤っていれば、指示文やエージェントの推論側（「[エージェント定義](#)」）を見直します。

困ったときの確認

状況	確認・対応
アップロードがエラーになる	形式とサイズがテキスト抽出設定の範囲内か確認する（「 取り込めるファイル 」）
登録したのに検索されない	ベクトル化ジョブが動作しているか確認する（「 ジョブの運用 」）。アイテムの状態が「待機中」のまま止まっていないかも確認する
チャンク分割設定を直したのに「エラー」のアイテムが復旧しない	エラー状態は設定変更だけでは自動的に「待機中」へ戻らない。原因解消後、アイテム一覧の「再ベクトル化」ボタンで再試行する（「 再ベクトル化のやり方 」）
「アイテム設定」画面で「処理中」のまま動かない	ベクトル化のジョブや「再ベクトル化」が実行中でないことを確認してから、「強制的に再ベクトル化待ちへ」ボタンで「待機中」に戻す。その後、ジョブの実行または「再ベクトル化」ボタンでベクトル化する（「 再ベクトル化のやり方 」）
IM-Wikiの更新が反映されない	外部ソース変更検知が有効になっているか確認する。既定では無効で、ソースタイプを選ぶだけでは自動追従しない。有効にしている場合は、「ナレッジアイテム外部ソース変更検知」のジョブが動いているかも確認する（「 ジョブネットの作成と有効化 」）
アイテム追加で IM-Wiki コンテンツの選択画面が開けない	操作するユーザに、IM-Knowledge の認可リソース「Knowledge コンテンツ利用者」の実行権限があるか確認する。「Accel Agent 管理者」ロールには含まれていない（「 同梱ロールが許すもの 」）
アイテムが急に検索対象から外れた	有効期間終了日を過ぎて「無効」かどうか確認する
登録したはずのアイテムが検索対象にならない	有効期間開始日が未来の日付かどうか（「開始待ち」状態）確認する
エージェントに割り当てたのに参照されない	エージェント定義のナレッジベースの割り当てで、Top-K・類似度閾値・メタデータフィルタの条件を確認する（「 ナレッジベース 」）
ナレッジベースをエクスポートできない	配下のアイテムに、待機中・処理中・エラー・メタデータ同期失敗のいずれかが 1 件でも含まれていないか確認する（「 エクスポートの注意点 」）
メタデータ項目を修正したい	項目名・型は登録後に変更できない。作り直しが必要かどうかを、利用箇所への影響とあわせて判断する

ツール

ツールは、エージェントが実行時に呼び出す実行部品です。組み込み・ロジック・MCP の 3 区分があり、区分ごとに登録・設定・利用箇所・削除の制約が異なります。

本章では、3 区分の使い分けから、それぞれの設定項目、副作用のあるツールの扱い方までを説明します。

項目

- ツールで確かめること
 - LLMに公開されるツール名の一意化
 - 画面を開く
- 組み込みツールを割り当てる
- ロジックツールを作る
- MCPサーバに接続する
 - OAuth の認証プロバイダを登録する
- MCPのセキュリティに注意する
- 副作用のあるツールの扱い方
- 困ったときの確認

ツールで確かめること

ツールの検討は、まず組み込みツールで足りるかどうかを確認するところから始めます。足りない場合に、業務固有の処理はロジックツールで、外部サービスへの接続は MCP で補う、という順序で検討します。

ロジックツールでは、ツールの「指示」（AI 向けの説明）が LLM の選択精度を左右します。何ができるかだけでなく、いつ呼ぶべきかを書きます。組み込みツールと MCP のツールには、指示を書く欄はありません。

LLMに公開されるツール名の一意化

エージェントに割り当てたツールは、区分を問わず LLM へ公開する際に 1 つの名前空間に並びます。複数のツールが同じ名前を名乗ろうとした場合、先に登録された側がそのまま名乗り、後から加わった側には 2 のような連番が自動で付きます。ツール名は 64 文字までで、連番を付けると上限を超える場合は元の名前を切り詰めます。したがって、割り当てたツールの呼び出し名が一覧の表示名と一致しないことがあります。実際に LLM へ渡る名前は、指示文やトレースで確認します。

画面を開く

サイトマップから「Copilot」→「Accel Agent」→「ツール管理」を開きます。画面は「組み込み」「ロジック」「MCP」の 3 タブに分かれています。タブごとに独立した一覧になっており、タブをまたぐ検索はできません。

組み込みツールを割り当てる

組み込みツールは、製品にあらかじめ用意されているツールです。エージェント定義の編集画面でツールとして選び、割り当てて利用します（「[ツール](#)」）。

組み込みツールは製品ごとに提供されており、一覧に表示されるツールは、導入しているモジュールによって変わります。例えば、IM-Workflow のモジュールを導入している環境では、ワークフロー関連のツールが表示されます。

一覧はカテゴリ・サブカテゴリで絞り込めるほか、ツール名と説明をキーワードで検索できます。ツールを選ぶと、そのツールの説明と、設定できる項目が表示されます。

設定できる項目は 2 種類です。

項目	値を決める人	値を決めるとき
入力パラメータ	LLM	ツールを実行するたび
プロパティ	エージェントの作成者	ツールを割り当てるとき

プロパティは LLM には公開されず、値を設定しなかった場合も LLM が値を決めることはありません。設定した値は、そのエージェントのすべての実行で共通に使われます。各項目の意味は、画面に表示される説明を参照してください。

動作を確認するには、提供元の製品側にデータが必要です。IM-Workflow のツールであれば、対象となるフロー定義や案件をあらかじめ用意してください。

ロジックツールを作る

ロジックツールは、IM-LogicDesigner のロジックフローを、業務固有のツールとしてエージェントに割り当てる機能です。登録する項目は、ツールID、ツール名、カテゴリ、説明（人が読む）、指示（AI 向けの説明）、参照するロジックフローとそのバージョンです。

項目	上限	備考
ツールID	64 文字	<code>new-flow-id</code> ・ <code>categories</code> は予約語のため使用不可（管理画面の REST パスと衝突するため）

項目	上限	備考
ツール名	255 文字	一覧・選択画面での表示名
説明	1000 文字	人が読む説明
指示	1000 文字	AI 向けの説明。LLM がこのツールを利用する際に参照する



図：ロジックツールの詳細画面。ツールID・指示・参照するロジックフロー・入力パラメータが並ぶ。

設計するときの勘所は、ツールを API の薄いラップにしないことです。曖昧な入力を許容する、パラメータを絞る、結果を構造化して返す、副作用を明示する、といった設計判断が必要です。例えば「案件ID を受け取ってフィールド値をそのまま返す」だけのツールにすると、LLM は返ってきた生データをどう解釈すべきか自分で判断しなければなりません。「在庫数」「単価」のように意味のわかる形に整えて返すと、後続の判断の精度が上がります。

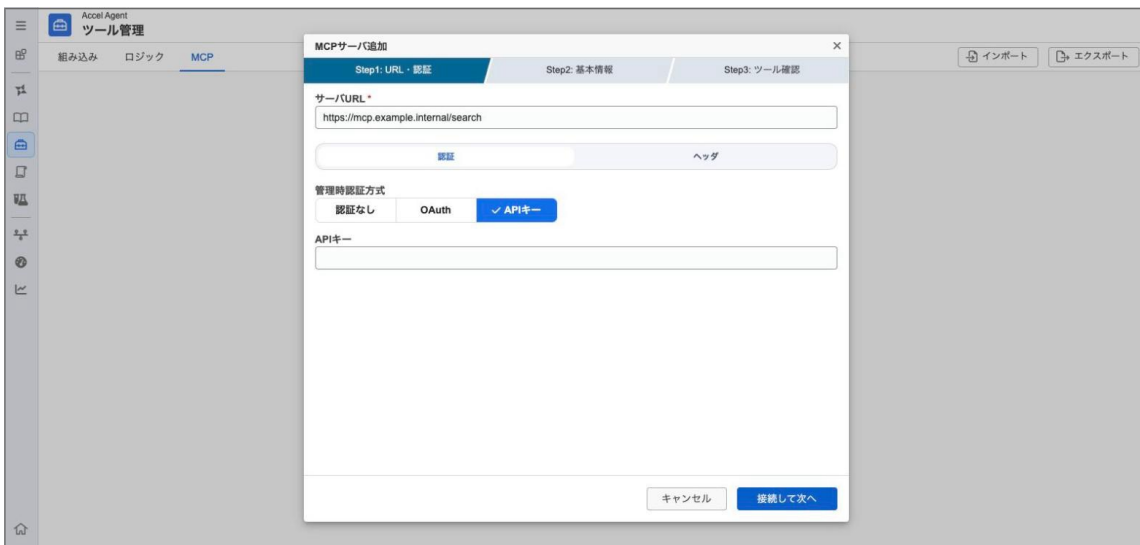
入出力の定義はロジックフロー側が正であり、ツール側に写しは持ちません。バージョンは「常に最新」を参照するか、特定のバージョンに固定するかを選べます。「常に最新」にすると、ロジックフロー側の入出力変更がエージェントの動作に無断で影響するため、頻繁に更新されるフローを参照する場合はバージョンを固定し、更新のたびに動作を確認してから切り替える運用を検討します。

ロジックフロー側には、実行制御用の予約パラメータ `imcoToolExecutionContext` が自動的に付与されるため、入力パラメータの定義でこの名前を使わないようにします。

MCPサーバに接続する

MCP は、外部サービスに接続する選択肢です。判断材料として最も重要なのは、intra-mart Accel Platform の権限コンテキストが引き継がれない点にあります。外部サービス側で誰の権限として動作するかを、別途設計する必要があります。

追加は 3 ステップのウィザードで進みます。



図：MCPサーバ追加ウィザードの Step1。サーバURLと、管理時認証方式（認証なし・OAuth・APIキー）を指定する。

ステップ	内容
Step1: URL・認証	サーバURLと管理時の認証方式を指定して接続する。追加の HTTP ヘッダも指定できる
Step2: 基本情報	サーバID・サーバ名・説明・タイムアウトと、実行時の認証方式を設定する。指示は、サーバが返す場合に表示される（編集不可）

ステップ	内容
Step3: ツール確認	接続先から取得したツール定義の一覧を確認する

認証は、なし・API キー・OAuth のいずれかを、管理時と実行時で個別に設定できます。OAuth には、ユーザの認可を得てアクセストークンを取得する認可コード（Authorization Code）と、ユーザ操作なしに JWT アサーション（RFC 7523）でアクセストークンを取得するクライアントクレデンシャルの 2 方式があります。管理者が事前に接続確認する場面では認可コードを、エージェント実行時にユーザ操作を挟めない場面ではクライアントクレデンシャルを選ぶ、という使い分けです。接続方式は Streamable HTTP のみです。プライベートアドレス帯のサーバは、既定の URL ブロックリスト設定では接続できません（「[MCPサーバのURL ブロックリスト設定](#)」）。

Step3 では、取得したツール定義への注意喚起が表示されます。これは「[MCPのセキュリティに注意する](#)」で説明する Tool Poisoning・サプライチェーン攻撃を指しており、ツール名と説明を目視で確認してから登録します。

ツール一覧は自動更新されず、更新しても差分は表示されません。人が目視で確認する運用が必要です。

OAuth の認証プロバイダを登録する

認証方式に OAuth を使う場合は、接続先の認可サーバの情報を「認証プロバイダ」として登録しておき、ウィザードで選びます。Step1 の管理時認証方式、または Step2 の実行時の認証方式で OAuth を選ぶと、「認証プロバイダを選択」ダイアログが開きます。使う認証プロバイダがまだない場合は、このダイアログの「新規作成」から登録します。

登録した認証プロバイダは、intra-mart Accel Platformの OAuth プロバイダ設定として保存されます。インポート・エクスポートの説明（「[インポート先の前提とインポートの順番](#)」）やロールの説明（「[同梱ロールが許すもの](#)」）にある「OAuth プロバイダ設定」は、ここで登録する内容です。

入力する項目は次のとおりです。

項目	内容
プロバイダID・名前・説明・アイコン	認証プロバイダを識別する情報を入力する
認証方式	認可コード、またはクライアントクレデンシャルを選ぶ。選んだ方式によって、入力する項目が変わる
プロバイダ種別	選ぶと、その種別のテンプレートで名前・エンドポイント・スコープなどの値が入力される
発行者識別子（issuer）	MCP サーバから認可サーバの情報を取得できた場合は自動で入力され、変更できない。取得できなかった場合は手入力する
トークンエンドポイント・クライアントID・スコープ・追加パラメータ	接続先の認可サーバに合わせて入力する
認可エンドポイント・クライアントシークレット・PKCE方式	認証方式が認可コードの場合だけ入力する
鍵形式（PEM・PKCS#12）・秘密鍵・署名アルゴリズムなど	認証方式がクライアントクレデンシャルの場合だけ入力する。JWT アサーションの署名に使う鍵を設定する

MCPのセキュリティに注意する

MCP には、Tool Poisoning、Tool Shadowing、Rug Pull、間接プロンプトインジェクションといった、この接続方式に固有のリスクがあります。

リスク	概要
Tool Poisoning	ツールの説明文に、LLM への指示（プロンプトインジェクション）を仕込む攻撃
Tool Shadowing	悪意のあるツールの説明で、他の正規ツールの動作を上書き・改変させる攻撃
Rug Pull	登録時は無害だったツール定義が、接続先の変更によって後から悪意ある内容に差し替わる攻撃
間接プロンプトインジェクション	ツールの実行結果（戻り値）に埋め込まれた指示を、LLM が本物の指示と誤認して従ってしまう攻撃

信頼できる MCP サーバだけを利用すること、追加時・更新確認時にツール名と説明を目視で確認することが、利用者側の基本です。ツール一覧が自動更新されない仕様（「[MCPサーバに接続する](#)」）は、裏を返せば Rug Pull を画面の自動反映では検知できないことを意味します。定期的な目視確認を運用に組み込みます。

副作用のあるツールの扱い方

読み取り専用のツールは自動実行してよいものの、作成・更新・外部送信を伴うツールは、許可の条件を満たす場合に限って実行させる、という考え方が一般的です。ただし、これは設計上の心得であって、製品機能として強制されるものではありません。

実行前に人の承認を挟む仕組みは、実装には存在しません。MCP のプロトコルが持つ「読み取り専用」といったヒントは受信していますが、実行制御には使っていません。実際に実行を止められるのは、ガードレールの検査だけです（「[ガードレール](#)」）。副作用のあるツールを扱うエージェントでは、ツール入力の検査位置にガードレールを置き、想定外の引数（例えば承認や送信を伴う操作）を検知したら止める、という組み合わせが現実的な防御策です。

困ったときの確認

状況	確認・対応
組み込みツールが一覧に出ない	供給元となる製品が対象バージョンに導入されているか確認する

状況	確認・対応
ロジックツールのID を登録できない	64文字以内か、予約語（new-flow-id・categories）を使っていないか確認する
複数のツールを割り当てたら呼び出し名がずれた	LLMに公開する名前は先着順で確定し、後発には連番が付く。実際の呼び出し名をトレースで確認する
MCPサーバに接続できない	プライベートアドレス帯のサーバは既定のURLブロックリストで接続できない（「 MCPサーバのURLブロックリスト設定 」）。管理時・実行時それぞれの認証設定も確認する
MCPの認証でユーザ操作が挟めない	クライアントクレデンシャル方式（JWTアサーション）を検討する。認可コード方式はユーザの認可操作が必要
MCPツールの一覧が古い	一覧は自動更新されない。差分も表示されないため、人が目視で確認する
ツールを削除できない	ロジックツール・MCP・スキルは、いずれかのエージェントに割り当てられている（利用中の）間は削除できない
実行時に利用者ID が必要になった	実行コンテキストからは取得できない。ロジックフロー側で解決する

スキル

スキルは、LLM に読ませる手順書です。Skill ID とスキル名、SKILL.md、利用者向け説明、補助ファイルをひとまとめにして管理します。

項目

- スキルで確かめること
 - 画面を開く
- スキルを構成する要素
- スキルを作る
- スキルが使われる仕組み
- エージェントへの割り当てと利用箇所の確認
- インポート・エクスポート
- 困ったときの確認

スキルで確かめること

スキルは、LLM に読ませる手順書であって、ツールのような実行部品ではありません。この前提を取り違えると、実行できるはずのない処理をスキルに書いてしまいます。手順を実際に実行させる必要がある場合は、スキル単体では完結しないため、ツールと組み合わせます（「[ツール](#)」）。

スキルにはカテゴリの区分がなく、一覧は検索だけで絞り込みます。ナレッジベースやツールのように業務ドメインで事前に整理する必要はなく、名前と説明で見分けられる粒度で作成して構いません。

画面を開く

サイトマップから「Copilot」→「Accel Agent」→「スキル管理」を開きます。一覧では、Skill ID とスキル名、説明の抜粋を確認できます。SKILL.md 自体やフォルダ名に問題があるスキルは「壊れています」と表示され、そのままでは利用できません。取り除いてから登録し直します。

スキルを構成する要素

スキルは、SKILL.md と補助ファイルをひとまとめにしたものです。ここでは、その中身が何で構成されるかを説明します。実際に作成する画面の操作は「[スキルを作る](#)」で扱います。

SKILL.md の先頭には、`---`で囲んだフロントマターとして `name`（画面の「スキル名」）と `description`（画面の「指示（AI用）」）という 2 つの項目を書きます。Skill ID はスキルを格納するフォルダの名前で、`name`とは別の値です。`---`が無いと、この部分はフロントマターとして認識されず、SKILL.md 全体が本文として扱われます。`description` は、LLM がこのスキルを選ぶかどうかを判断する唯一の手がかりです。何ができるかだけでなく、いつ使うべきかをここに書きます。

例えば、次のように書きます。

```
---
name: ticket-triage
description: 問い合わせメールの内容から緊急度を判定し、担当チームへの振り分け案を作るときに使う
---
```

（ここから本文。LLM に読ませる具体的な手順を書く）

このフロントマターは、フル仕様の YAML ではなく専用の簡易パーサで読み取ります。`key: value`の単純な書き方と、説明文を複数行に分けたいときの `description: |`（改行を残す）のようなブロックスカラ、`metadata:`配下に1段だけ `key: value`を並べる書き方に対応しています。フロー形式（`[a, b]`）や2段以上のネストなど、凝った書き方はできません。構文がおかしいと取り込みに失敗します（「[困ったときの確認](#)」）。

項目	上限	用途
Skill ID	64 文字	小文字英数字とハイフンのみ（先頭・末尾・連続ハイフン不可）。例: <code>ticket-triage</code>
<code>name</code> （スキル名）	64 文字	Skill ID と同じ形式。Skill ID と一致させる必要はない
<code>description</code>	1024 文字	LLM 向け。いつこのスキルを使うべきかを書く
画面の項目	良い例	悪い例
指示（AI用）	問い合わせメールの内容から緊急度を判定し、担当チームへの振り分け案を作るときに使う	問い合わせ対応に関するスキル

悪い例は対象を述べているだけで、いつ使うべきかが書かれていません。複数のスキルを割り当てたエージェントでは、この違いが実際に選ばれるかどうかを左右します。

SKILL.md 本文（フロントマターより後ろの部分）には、手順そのものを書きます。ここは LLM 向けの「指示」であり、人が読む利用者向け説明とは別です。利用者向け説明は SKILL.md の中ではなく、SKILL.md と同じ場所に置く予約ファイル `.description`（ローカル別は `.description_<ローカル>`）に書きます。作成画面の「説明」欄に入力した文章はこの `.description`ファイルに保存され、Skill 一覧に表示されるだけで、LLM の選択には使われません。SKILL.md と説明ファイルを混同すると、人向けの文章が LLM の選択材料に混ざってしまいます。

編集画面では、SKILL.md 以外の補助ファイルを追加・リネーム・移動できます。ファイル追加時はファイル名とフォルダ名を、リネーム時は新しい名前を、移動時は移動先フォルダを指定します。補助ファイル（SKILL.md 以外にスキルへ追加できるファイル）には上限があります。

制限	既定値
1 ファイルの最大サイズ	256KB
1 スキルが持てるファイルの最大数	500
取り込む ZIP 全体の最大サイズ（展開後の合計）	100MB

上限を超えると、切り詰めずに失敗します。登録・インポートの入口では取り込み自体を拒み、実行時にはツールの実行を失敗させます。一部だけを渡すと、LLM が本文全体を読んだものとして続きを組み立ててしまうためです。SKILL.md は UTF-8 で書きます。UTF-8 として読めない場合は、文字化けしたまま取り込まれることを避けるため、取り込みを拒みます。

下位フォルダに置いた SKILL.md（入れ子スキル）は対応していません。スキルの定義になるのは、直下に置いた SKILL.md 1 個だけです。

スキルを作る

作成の経路は、空から書く、ZIP を取り込む、AI に生成させるの 3 通りがあります。いずれも作成方法を選ぶ画面から始まり、空から書く場合はそのまま基本情報（Skill ID・スキル名・指示・説明）の入力画面に進みます。

経路	進み方
空から作成	SKILL.md だけを含む空のワークスペースから編集を始める
ZIP をアップロードして作成	SKILL.md を含む既存の Skill バンドル（ZIP）をインポートして編集を始める。ドラッグ&ドロップにも対応
AI で生成	やりたいことをプロンプトで伝え、AI に Skill の雛形を生成させる



図：Skill の作成方法を選択する画面。空から作成・ZIP をアップロードして作成・AI で生成の 3 通りから選ぶ。

作成画面の入力欄は、Skill ID・スキル名・指示（AI用）・説明の 4 つです。このうちスキル名と指示（AI用）が、そのまま SKILL.md の `name`・`description` に書き込まれます。画面上は「指示（AI用）」という名前でも、実は SKILL.md の `description` フィールドである点に注意します。



図：空から作成を選んだ場合の基本情報入力画面。Skill ID・スキル名・指示（AI用）・説明を入力する。

AI 生成では、基本情報の入力後に生成用プロンプトを追加で入力します。生成中は、SKILL.md の生成、補助ファイルの生成、AI への問い合わせ（複数回になること

があります)といった進行状況が画面に表示されます。ツール呼び出しが失敗した場合は自動的にやり直されます。生成後の中身はたたき台として扱い、実際に使う前に SKILL.md と補助ファイルの内容を確認します。AI に生成させる場合は、暴走を止めるための上限として、1 ファイル 200KB・最大 30 件までという、「[スキルを構成する要素](#)」の上限より厳しい枠が別に設けられています。

ZIP を取り込む際、Skill ID には予約語が使えません (list・draft・export・import・generate・catalog・limits)。これらは管理画面の REST パスで固定語として使われているため、スキルID に許すと、パスを見ただけではスキルを指すのか操作を指すのか区別できなくなります。

ZIP の展開時、macOS の .MACOSX・.DS_Store や Windows の Thumbs.db・desktop.ini といった OS の作業ファイルは、利用者が入れたものではないため無かったものとして扱われます。大文字・小文字だけが違う同名パスが 2 つ以上あると、取り込みを拒みます (大文字小文字を区別しないファイルシステムでは、後から書いたほうだけが残し、片方が黙って消えるためです)。

スキルが使われる仕組み

スキルの中身は、概要 (Skill ID・name・description・metadata) だけを先に読み込み、必要になった時点で本文を取得する段階取得の仕組みで扱われます。一覧・選定の段階では本文を読み込まないため、長い手順書を書いても、選定時のコンテキストを圧迫しません。本文は、実際にそのスキルが使われる段になって初めて取得されます。

この仕組みが、「[スキルを構成する要素](#)」で説明した description が選定の唯一の手がかりになる理由です。選定の時点ではまだ本文が読み込まれていないため、本文にどれだけ詳しく書いても、それだけでは選ばれません。

エージェントへの割り当てと利用箇所の確認

作成したスキルは、エージェント定義の画面から割り当てます (「[ナレッジベース・スキル・ツール・ガードレールを割り当てる](#)」)。割り当てたスキルの利用箇所は、スキル管理画面側からも確認できます。利用中 (いずれかのエージェントに割り当てられている状態) のスキルは削除できません。

インポート・エクスポート

スキルは ZIP 形式でインポート・エクスポートできます。手順と注意事項は「[インポート／エクスポート](#)」を参照してください。

困ったときの確認

状況	確認・対応
スキルの取り込みが失敗する	ファイルサイズ・件数の上限を超えていないか、SKILL.md が UTF-8 で書かれているか確認する
SKILL.md のフロントマターが読み込めない	---の開始・終了が揃っているか、引用符が閉じているか、対応する項目の無い字下げ行がないか確認する。フル YAML ではないため、フロー形式 ([a, b]) や2段以上のネストは使えない
Skill ID が登録できない	64 文字以内で、小文字英数字とハイフンのみか、予約語 (list・draft・export・import・generate・catalog・limits) を使っていないか確認する
LLMがスキルを選ばない	画面の「指示 (AI用)」=SKILL.md の description に、いつ使うべきかが書かれているか確認する。対象を述べるだけでは選ばれない
利用者向け説明が反映されない	SKILL.md 本文ではなく、作成画面の「説明」欄 (保存先は .description ファイル) に書かれているか確認する
一覧に「壊れています」と出る	SKILL.md 本体がフォルダ名に問題がある。取り除いてから登録し直す
スキルを削除できない	いずれかのエージェントに割り当てられている (利用中の) 間は削除できない。利用箇所を先に確認する
AI生成が途中で止まる、または内容がおかしい	生成結果はたたき台であり、そのまま使わず SKILL.md と補助ファイルを確認する。ツール呼び出し失敗時は自動でやり直されるが、生成プロンプトが曖昧だと期待と異なる雛形になる

エージェント定義

エージェント定義は、ナレッジベース・ツール・スキル・ガードレールという材料を組み立てて、1つのエージェントに仕立てる画面です。「[本格構築を始める前に](#)」で決めた5点をもとに、9つのセクションを順に埋めていきます。

項目

- 画面を開く・全体の流れ
- 基本情報・モデル設定を決める
 - 基本情報
 - モデル設定
- 指示を書く
- 入出力値を設計する
- ナレッジベース・スキル・ツール・ガードレールを割り当てる
 - ナレッジベース
 - スキル
 - ツール
 - ガードレール
- 実行制御を設定する
- テスト実行で確認する
- バージョンを管理し、公開して改善する
- 困ったときの確認

画面を開く・全体の流れ

サイトマップから「Copilot」→「Accel Agent」→「エージェント管理」を開きます。一覧から既存のエージェントを開くか、新規作成すると、編集画面に入ります。エージェント定義の編集画面には、左ナビゲーションに並ぶ9つの設定セクションと、バージョン・テスト実行に関する2つの運用機能があります。

基本情報・モデル設定を決める

基本情報

基本情報は、エージェントを一意に識別し、検索・分類できるようにするための項目です。保存には、エージェントID（新規作成時は自動採番）・エージェント名・カテゴリ・AIモデルが必須で、これらが未入力の場合は保存できません。説明は任意項目です。

用途は「ロジックフロータスク」「BPMタスク」から複数選択できます。BPMタスクは、IM-BPMが未導入の環境では選択肢に表示されません。用途は呼び出し元を絞る検索条件であり、実行時のアクセス制御ではない点に注意します（「[エージェントの呼び出し方](#)」）。

モデル設定

モデル設定では、エージェントが呼び出すLLMと、その挙動パラメータを指定します。常に必須で、未選択のままではテスト実行・公開実行のいずれもできません。

AIモデルは、「[AIモデル管理](#)」にあらかじめ登録したAIモデルのうち、テキスト生成用のもの（埋め込み用は選択肢に出ません）を1つ選びます。選ぶと、ドライバ種別・プロバイダモデルIDが画面に補足表示されます。

調整できるパラメータは、ランダム性（temperature）と最大出力トークン数（max_tokens）の2つです。AIモデル管理で有効としたパラメータだけが欄に表示され、指定できる範囲もその登録内容で決まります。

指示を書く

指示は、エージェントの役割や守るべきルール、回答の形式などを定めるシステムプロンプトです。エージェントがどのように判断し、振る舞うかの基本方針となり、LLMは通常、そのつど渡される質問や依頼よりも、この指示を優先して処理します。

指示には、テンプレート変数を埋め込みます。テンプレート変数は、インプットパラメータの値や実行したユーザの情報など、実行のたびに変わる値を指示の中に差し込むための仕組みです。`${変数名}`の形式で書くと、実行時に実際の値へ置き換えてからLLMに渡されます。例えば、インプットパラメータ `quoteText` を定義して指示に `${quoteText}` と書くと、その箇所に呼び出し元から渡された値が入ります。IM-LogicDesignerの「エージェント実行」タスクから呼び出す場合、この値はタスクのマッピング設定で `inputParameters` に割り当てた値です（「[IM-LogicDesigner の「エージェント実行」タスク](#)」）。入力欄で `${` と入力すると変数の候補が表示され、「コンテキスト変数を挿入」からはクリックでカーソル位置に挿入できます。

テンプレート変数の供給元は、トップレベルのインプットパラメータと、次のコンテキスト変数です。

分類	参照できる値
アカウント情報	現在の操作対象のテナントID、現在アクセスしているユーザのユーザコード、ユーザ種別、ログイン認証済みかどうか、ユーザが利用するロケール・タイムゾーン、ログイン時刻、ユーザのロールID一覧（カンマ区切り）、文字エンコーディング、ユーザが利用するホームURL
ユーザ情報	ログインユーザの表示名、ユーザのメールアドレス、カレント組織の会社コード、カレント組織コード、カレント組織名

Accel Agent 利用ガイド — Accel Agent 利用ガイド 初版 2026-10-01

呼び出しのたびに変わらない値は、テンプレート変数にせず、指示の本文に直接書きます。コンテキスト変数から取れるのは、テナントID やユーザコードのような実行したユーザに関わる値だけです。案件の内容などの業務データは、呼び出し元からインプットパラメータとして渡すか、実行中に取得する必要がある場合はツールの戻り値として受け取る設計にします。

変数名を書き誤った場合や、存在しない変数を書いた場合は、エラーにならず、`${quoteTxt}` のように書いたままの文字列が LLM に渡されます。実行はそのまま続くため、値が置き換わっていないことに気づきにくい点に注意します。一方、`$` の閉じ忘れなど、テンプレートの書き方そのものに誤りがある場合は、実行が始まる前にエラーが発生します。

入出力値を設計する

入出力値は、エージェントへの入力と、エージェントからの出力の形式を定義します。インプットパラメータは呼び出し元から値を渡したい場合に、構造化出力は出力を後続の処理で機械的に扱いたい場合に設定します。どちらも設定しない場合、入力は自然文の **prompts** のみになり、出力は「テキスト出力」（LLM の応答をそのまま文字列で返す）です。

出力は「テキスト出力」と「構造化出力 (JSON)」をトグルで切り替えます。構造化出力を ON にすると、出力パラメータとして JSON スキーマ相当の定義（パラメータ名・型・説明）を作成でき、LLM の応答をその形式に合わせて返させます。後続の処理で値を機械的に使う予定があるなら、自然文ではなく構造化出力にします。例えば「勘定科目の候補・理由文・照合結果」をロジックフロー側でそれぞれ別の項目として扱いたいなら、この 3 つを出力パラメータとして定義し、構造化出力を ON にします。

図：入出力値の設定画面。インプットパラメータの定義と、構造化出力 ON 時の出力パラメータ定義が並ぶ。

ナレッジベース・スキル・ツール・ガードレールを割り当てる

判断に使う情報や実行手段は、「[ナレッジベース](#)」・「[スキル](#)」・「[ツール](#)」のいずれかで持たせます。安全のための検査は「[ガードレール](#)」で設定します。それぞれの設計・作成は各章で扱うため、本節ではエージェントへの割り当て方だけを扱います。

ナレッジベース

ナレッジベースを割り当てると、エージェントに意味検索で参照できる情報源を持たせられます。画面上、Top-K と類似度閾値は未設定の状態が既定で、この場合は実行時の既定値である Top-K 5、閾値なしが適用されます。

割り当てたナレッジベースごとに、メタデータフィルタを組み合わせで検索範囲を絞り込みます。条件の値は、固定値（静的指定）と、インプットパラメータへの参照（動的指定）のどちらかで指定でき、動的指定にすると実行のたびに呼び出し元が渡した値で条件が決まります。参照先のインプットパラメータが未指定のとき、その条件を無視するか検索を止めてエラーにするかも条件ごとに選べます。



図：メタデータフィルタの設定例。値の参照方式を「入力パラメータ」（動的指定）にし、参照先のインプットパラメータを選んだ状態。

メタデータフィルタで参照できる項目は、そのナレッジベースに定義されたメタデータ項目に限られます（「[メタデータ項目を設計する](#)」）。プロパティを指定した精度評価では回答関連性を算出しません点にも注意します（「[データセットを確認・修正する](#)」）。



図：ナレッジベースの割り当て画面。割り当て済みのナレッジベースが一覧で並ぶ。

スキル

スキルを割り当てると、エージェントに実行時参照させる手順書を持たせられます。割り当てたスキルの利用箇所は、スキル管理画面側からも確認できます（「[エージェントへの割り当てと利用箇所の確認](#)」）。

ツール

ツールを割り当てると、エージェントに組み込み・ロジック・MCP のツールを持たせられます。割り当てたツールが後から削除・改名された場合の挙動は、ツールの区分によって異なります（「[ツール](#)」）。参照が切れた状態のまま実行すると、該当ツールの呼び出しが失敗します。



図：ツールの割り当て画面。組み込み・ロジック・MCP を種別ごとに設定する。

ガードレール

ガードレールを設定すると、エージェントの入出力を検査し、必要に応じて実行を止める、または記録できます。詳細な設定項目・値の範囲は「[ガードレール](#)」にまとめています。

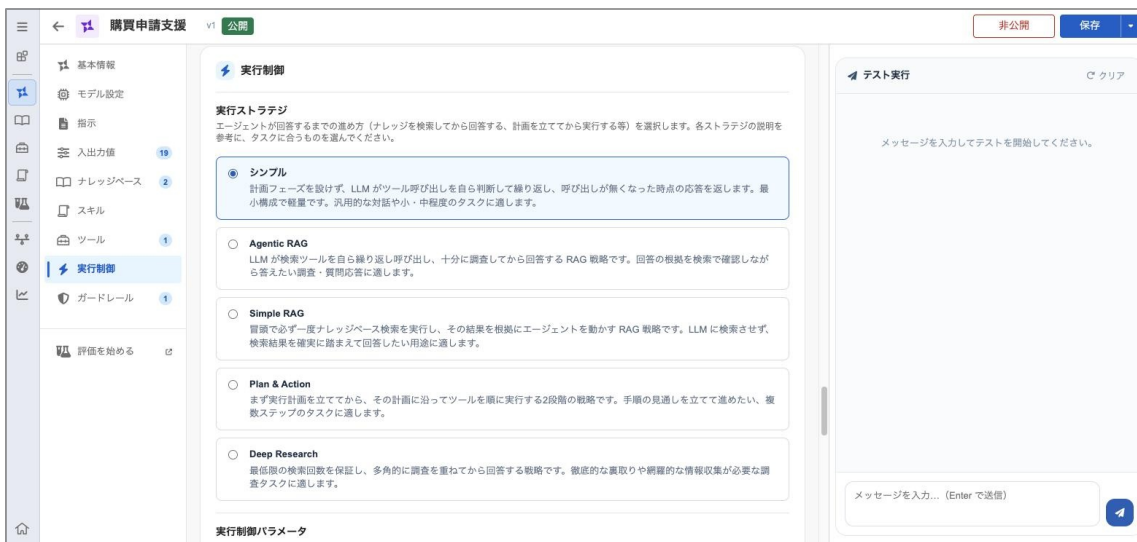
実行制御を設定する

実行制御では、エージェントの実行方式と、実行の上限を定義します。ストラテジは常に選択が必要ですが、最大ループ回数・タイムアウト・自動コンテキスト圧縮の発動割合はいずれも任意項目です。最大ループ回数を未設定にすると既定値の 100 回が適用され、タイムアウトと自動コンテキスト圧縮の発動割合は未設定の場合は無効として扱われます。

！ 注意

自動コンテキスト圧縮の発動割合が未設定の場合、会話履歴は圧縮されずに蓄積されます。会話履歴を含むリクエストのトークン数が AI モデルのコンテキストウィンドウを超えると、AI サービスからエラーが返り、実行は中断されます。同じセッションで会話を続ける使い方をする場合は、発動割合を設定してください。

発動割合を設定していても、AI モデル管理でコンテキストウィンドウを登録していないモデルでは圧縮は行われません。ロジックフローから呼び出す場合の条件は「[IM-LogicDesigner の「エージェント実行」タスク](#)」を参照してください。



図：実行制御のストラテジ選択。ストラテジごとの動き方と向いている用途が説明文で示される。

ストラテジには、次のものがあります。タイムアウトはミリ秒単位で指定します。

ストラテジ	動き方	向いている用途
シンプル	計画フェーズを設けず、LLM がツール呼び出しを自ら判断して繰り返し、呼び出しがなくなった時点の応答を返す。最小構成で軽量	汎用的な対話や小・中程度のタスク
Agentic RAG	LLM が検索ツールを自ら繰り返し呼び出し、十分に調査してから回答する RAG 戦略	回答の根拠を検索で確認しながら答えたい調査・質問応答

ストラテジ	動き方	向いている用途
Simple RAG	冒頭で必ず一度ナレッジベース検索を実行し、その結果を根拠にエージェントを動かす RAG 戦略。LLM に検索させない	検索結果を確実に踏まえて回答したい用途
Plan & Action	まず実行計画を立ててから、その計画に沿ってツールを順に実行する 2 段階の戦略	手順の見通しを立てて進めたい、複数ステップのタスク
Deep Research	最低限の検索回数を保証し、多角的に調査を重ねてから回答する戦略	徹底的な裏取りや網羅的な情報収集が必要な調査タスク

Simple RAG には、検索前にユーザ入力を検索に適したクエリへ自動で書き換える「クエリ書き換え」パラメータがあります。Deep Research には「最低検索回数」パラメータがあり、回答前に最低限行うナレッジベース検索の回数を指定します。多いほど網羅的に調査しますが実行時間は増え、最大ループ回数が上限です。



図：Plan & Action を選んだ状態。選択したストラテジに応じて、実行制御パラメータの構成が変わる。

テスト実行で確認する

テスト実行は、保存前・公開前のエージェント定義の動作を確認する機能です。編集画面右側のドックから実行でき、応答本文のほか、トークン数と所要時間があわせて表示されます。構造化出力を ON にしている場合は、構造化出力の結果も確認できます。

セクションを埋めるたびにこまめにテスト実行すると、どの変更が挙動を変えたかを切り分けやすくなります。特に指示文とナレッジベース・ツールの割り当ては相互に影響するため、これらを変更したときは都度確認します。テスト実行の記録は、公開後の本番実行の記録とは区別される点にも注意します。

バージョンを管理し、公開して改善する

バージョン管理・公開の機能では、エージェント定義のバージョンを管理し、呼び出し元に公開するバージョンを制御します。公開中のバージョンを直接上書き保存すると、変更は呼び出し元へ即座に反映されます。この性質があるため、改善は「新しいバージョンとして保存」してから精度評価（「[精度評価](#)」）を行い、問題がなければ公開する、という順序で回します。新しいバージョンをいつ作るかは、この即時反映の影響範囲をもとに判断します。公開中のバージョンと、最後に残った 1 つのバージョンは削除できません。



図：バージョン一覧。各バージョンの番号、公開状態、最新かどうかが並ぶ。

業務に組み込む具体的な手順は「[業務に組み込む](#)」、別環境で作った定義を持ち込む場合の手順は「[インポート／エクスポート](#)」を参照してください。

困ったときの確認

状況	確認・対応
テスト実行・公開実行ができない	AIモデルが選択されているか確認する
用途で絞ったのに呼び出せる／呼び出せない	用途は検索条件であり、実行時のアクセス制御ではない。アクセス制御は別途確認する
テンプレート変数が展開されず、 <code>\${...}</code> のまま LLM に渡される	変数名に誤りがないか、トップレベルのインプットパラメータかコンテキスト変数（アカウント情報・ユーザ情報）のいずれかを参照しているか確認する。存在しない変数はエラーにならず、書いたままの文字列で渡される。固定値は変数にできない
実行が始まる前にエラーになる	指示のテンプレートの書き方に誤りがないか確認する。 <code>`\${`</code> の閉じ忘れなどがあると実行できない。AIモデルが選択されているかも確認する
後続処理でJSONとして受け取れない	出力に「テキスト出力」が選ばれていないか確認する。「構造化出力」をONにして出力パラメータを定義する
割り当てたツールの呼び出しが失敗する	ツールが削除・改名されて参照が切れていないか確認する
バージョンを削除できない	公開中のバージョンと、最後に残った 1 つのバージョンは削除できない
変更したのに呼び出し元に反映されない、または意図せず反映された	公開中のバージョンを直接上書き保存すると即座に反映される。新しいバージョンとして保存してから精度評価を行い、問題がなければ公開する

ガードレール

ガードレールは、エージェントの入出力を検査し、必要に応じて実行を止める、または記録する仕組みです。エージェント定義の中の 1 セクションですが、検査位置・動作・判定スクリプトの組み合わせを設計する必要があります。本章の順に、何を検知したいかを決めてから、検査位置と動作を選びます。

項目

- [ガードレールで確かめること](#)
- [検査位置を選ぶ](#)
- [動作を選ぶ](#)
- [判定スクリプトを書く](#)
 - [runの引数](#)
 - [返り値の書き方](#)
 - [書き方の例](#)
 - [失敗したときの扱い](#)
- [逐次応答の検査設定](#)
- [保存前にテストする](#)
- [発動状況を確認する](#)
- [困ったときの確認](#)

ガードレールで確かめること

ガードレールの設計は、何を検知したいかを決めてから検査位置を選ぶ、という順番で進めます。検査位置によって検査できる対象が決まっているためです。検知したときの動作（止める／記録する）は、そのあとに決めます。すべてを塞ごうとするのではなく、実際に起きうる失敗（規程にない金額の送信、機微情報の出力など）から逆算して、必要な検査位置と動作を選びます。

設定はエージェント定義の「ガードレール」セクションに、検査位置ごとに追加していきます。1 件ずつ有効・無効を切り替えられるため、動作を確かめたい設定だけを残して他を一時的に無効にする、という調整もできます。

検査位置を選ぶ

検査位置は、入力・ツール入力・ツール出力・出力の 4 か所から選びます。同じ入出力でも、検査するタイミングによって止められるものが異なります。

検査位置	いつ検査するか	主な用途
入力	利用者のメッセージを、会話履歴へ追加する前に検査する	AI モデルへ送る前の内容を確認したい場合
ツール入力	ツールへ渡す引数（JSON 文字列）を、ツールを動かす前に検査する	外部への送信・実行を止めたい場合
ツール出力	ツールが返した値を検査する	ツールが返した内容に機微情報が含まれていないか確認したい場合
出力	利用者へ返す応答を検査する。逐次応答では、一定の文字数ずつ区切って検査してから送出する	最終的な回答の内容を確認したい場合

外部への送信を止めたいならツール入力に、AI モデルへ送る前の内容を確認したいなら入力に置く、というのが基本的な選び方です。同じ判定スクリプトでも、どの検査位置に置くかによって `ctx.tool`（ツール名。ツール入力・ツール出力以外の位置では空文字）などの値が変わるため、複数の検査位置で使い回す場合は、この違いを踏まえてスクリプトを書きます。

ツール入力・ツール出力フィルタが検査するのは、通常のツール呼び出しだけです。ナレッジ検索・スキルの読み込みは別の経路で実行されるため、この 2 つの検査位置では検査されません。また、ツール出力フィルタが検査するのは、戻り値の先頭 100,000 文字までです。超えた部分に機微情報があっても、このフィルタでは検知できません。戻り値そのものは切り詰められず、全文が LLM へ渡ります。大きな戻り値を返すツールを検査したい場合は、この上限を踏まえて設計します。

応答の返し方には、実行完了後に結果をまとめて返す「同期実行」と、生成中の文字を順に送る「逐次応答」があり、どちらになるかはエージェントの呼び出し方によって決まります。例えば、ロジックフローの「エージェント実行」タスクは同期実行で、エージェント編集画面のテスト実行は逐次応答です。ガードレールの設定で切り替える項目はありません。

出力フィルタが検査する範囲は、応答の返し方によって異なります。

- 同期実行では、ツール呼び出しを含まない最終応答だけを検査します。LLM がツールを呼び出したラウンドの応答（途中経過）は検査されません。
- 逐次応答では、途中経過を含め、利用者へ送る文字をすべて検査します。送出前にウィンドウ単位で検査し、通過した分だけ送出します。

構造化出力を使う場合は、整形した結果を別に検査します。同期実行では、整形前の本文は検査されません。逐次応答では、整形前の本文も送出前に検査します。

動作を選ぶ

動作には、止める（BLOCK）・記録する（LOG）・何もしない（PASS）の 3 択があります。どの検査位置でも共通です。最初から止めるのではなく、まず記録するで様子を見てから、必要に応じて止めるへ引き上げる運用を推奨します。

動作	実行への影響
止める（BLOCK）	実行を中止する

動作	実行への影響
記録する（LOG）	止めずに実行を続ける。発動はトレースに残る
何もしない（PASS）	何も起きない

ただし、記録する・何もしないの判定結果は監査ログには残りません。監査ログに記録されるのは、止めると判定した場合だけです。記録するの件数を確認したい場合は、トレース（「[実行を追う（トレース）](#)」）でエージェントごとの検査結果を見ます。

ツール入力・ツール出力の検査位置で止めた場合も、そのツール呼び出しだけでなく、実行全体が中止されます。1 回の応答で複数のツールを呼び出している途中で同様です。

同一の検査位置に複数のガードレールを設定した場合、登録順に評価されます。いずれかが止めると判定した時点で打ち切り、それ以降は評価されません。

判定スクリプトを書く

判定は、`function run(input, ctx) { ... }` の形式で書く判定スクリプトによって行います。構文はあらかじめ検証されないため、書いたら必ずテストします（「[保存前にテストする](#)」）。

判定スクリプトは、IM-LogicDesigner のユーザ定義スクリプトと同じ処理系・同じ API で動きます。テナント DB の参照・更新、ストレージへのアクセス、メール送信、外部通信など、業務用スクリプトと同等の操作が行え、実行時間の上限もありません（上限があるのは命令数だけです。「[失敗したときの扱い](#)」）。判定スクリプトは検査のためだけに使い、副作用のある操作は書かないようにします。

run の引数

引数	項目	内容
<code>input</code>	（検査対象の文字列）	検査位置ごとの対象。入力・出力はメッセージ本文、ツール入力は引数の JSON 文字列、ツール出力はツールの戻り値
<code>ctx</code>	<code>phase</code>	検査位置（ <code>INPUT</code> ・ <code>TOOL_CALL</code> ・ <code>TOOL_RESULT</code> ・ <code>OUTPUT</code> ）
	<code>tool</code>	ツール名（LLM に公開している表示名）。ツール入力・ツール出力以外の検査位置では空文字
	<code>window</code>	AI モデルの呼び出しごとに 1 から数えるウィンドウ番号
	<code>last</code>	その呼び出しの最後のウィンドウかどうか（実行全体の最後とは限らない）
	<code>state</code>	実行 1 回の間だけ持ち回れる入れ物

`ctx` 自体は書き換えられません。回数を数える、前のウィンドウの判定結果を覚えておくといった値の持ち回りは `ctx.state` に入れます。関数の外（トップレベル）に置いた定数・補助関数は、実行 1 回につき 1 度だけ評価されます。グローバル変数によって複数の実行をまたいで値を持ち越すことは保証されません。

`ctx.tool` に入る名前は、ツール名が重複して連番が付いた場合や、ツール定義側で名前を変更した場合に変わります（「[LLM に公開されるツール名の一意化](#)」）。`ctx.tool` で特定のツールだけを検査する判定を書く場合は、この名前が変わりうることを踏まえます。

逐次応答（ストリーミング）では、送出する本文のウィンドウ 1 つ分の文字列が `input` に入ります。構造化出力を使う場合、整形した結果の検査では、その結果が `input` に入って 1 回だけ呼ばれます。

逐次応答と構造化出力を両方使う場合、本文分のウィンドウ呼び出し（`window` を 1 から数える）に続けて、整形結果分の呼び出しが来ますが、整形結果分も `window` は 1 から数え直されます。どちらも同じ `ctx.state` を引き継ぐため、`window` が 1 のときに毎回初期化するような書き方をすると、本文分と整形結果分の数え上げが混ざります。

返り値の書き方

スクリプトは、`{action: "BLOCK"|"LOG"|"PASS", message: "..."}` の形をしたオブジェクトを `return` します。何も返さない（`undefined`）場合は「何もしない」と同じ扱いです。`message` には、止めるときに利用者へ提示する文言を入れます。止めるときに `message` を書かないと、利用者には何も表示されません。理由を伝えたい場合は必ず `message` を書きます。文言に、検知した内容そのものを書かないでください（検知した機密情報を `message` でそのまま返すと、検査の目的と矛盾します）。

書き方の例

画面には、次のような組み合わせの例（スニペット）が用意されています。実際の判定スクリプトは、これらの「見つけ方」と「返し方」を組み合わせで書きます。

分類	例
返し方	止めて文言を出す／文言を出さずに止める／止めずに記録だけする／何もしない（何も返さない）
見つけ方	語の一覧で探す／大文字と小文字を区別せずに探す／書式で探す（電話番号・メールアドレス・12 桁の番号）／長さで見ると JSON を読んで項目を見る／語の数を数える
ctx の使い方	特定のツールだけ検査する／回数を持ち回る（ <code>ctx.state</code> ）／最後のウィンドウでまとめて判定する／最初のウィンドウだけ検査する
関数の外	定数を置く／補助関数を置く

例えば、入力フィルタで「機密」「社外秘」という語を含む場合に止める判定スクリプトは、次のような形です。

```
function run(input, ctx) {
  var words = ["機密", "社外秘"];
  for (var i = 0; i < words.length; i++) {
    if (input.indexOf(words[i]) !== -1) {
      return { action: "BLOCK", message: "社外秘の語が含まれています" };
    }
  }
  return { action: "PASS" };
}
```

同様に、ツール入力フィルタでは「引数に外部の宛先があれば中止する」、ツール出力フィルタでは「戻り値が一定の長さを超えたら記録する」、出力フィルタでは「電話番号の書式があれば止める」といった判定が典型例として用意されています。いずれも、検査対象（input）から機械的に判定できる条件に絞り、曖昧な意味判断はさせない書き方になっている点が共通しています。

失敗したときの扱い

状況	実際の動作
スクリプトが例外を送出した	続行（何もしない）として扱う。警告ログに残る
命令数が上限に達した	続行（何もしない）として扱う。警告ログに残る
戻り値の書き間違い（run関数が無い、戻り値がオブジェクトでない、actionが無い、知らない動作名を返した、のいずれか）	続行（何もしない）として扱う。警告ログに残る

判定スクリプトの失敗は、どの経路でも実行を止めません。失敗に気づくには、まずトレース詳細でガードレールのスパンを確認します。判定が失敗した検査は、判定結果が「判定失敗」と表示されるため、どの実行のどの検査位置で失敗したかがわかります（「[実行を追う（トレース）](#)」）。

失敗の原因（スクリプトの例外の内容など）は、サーバのシステムログに警告として出力されます。判定の実行に失敗した場合はW.IWP.COPILOT.AGENT.GUARDRAIL.00001、命令数の上限に達した場合はW.IWP.COPILOT.AGENT.GUARDRAIL.00002のメッセージコードで、ガードレールの識別子（guardrailId）と検査位置（phase）とともに記録されます。これらを手がかりにログを検索します。スクリプトの書き間違いは、保存前のテスト（「[保存前にテストする](#)」）でも確認できます。

設定 1 件ごとに、識別子（200 文字以内・重複不可）、名称（255 文字以内、ロケール別も可）、説明（1000 文字以内、ロケール別も可）を持ちます。識別子・名称・検査位置・判定スクリプトは必須です。

逐次応答の検査設定

逐次応答（ストリーミング）では、出力を一定の文字数ずつに区切って検査してから利用者へ送出します。区切り方はエージェントに 1 組だけの設定です（検査位置ごとではありません）。

ウィンドウサイズ（一度に検査する文字数）は 64～8192 文字の範囲で指定でき、既定値は 512 です。オーバーラップ（区切り目をまたぐ語を見落とさない量）は 0 以上、かつウィンドウサイズの半分（小数切り捨て）以下で指定でき、既定値は 128 です。この上限を超える値は保存できません。

ウィンドウサイズ分の文字が溜まるまでは検査も送出も行われないため、ウィンドウサイズを大きくするほど、最初の応答が画面に表示されるまでの時間は長くなります。

見つけたい語や句がオーバーラップの文字数より長いと、区切り目をまたいだときに検出できないことがあります。長い語句を確実に検知したい場合は、オーバーラップを広げます。

オーバーラップの部分は、前後 2 つのウィンドウの両方で検査されます。この範囲に記録する（LOG）条件に合う語があると、発動が 2 回記録されることがあります。

逐次応答では、あるウィンドウで止めると判定しても、それより前のウィンドウはすでに利用者の画面に送信済みです。この送信済みの内容を取り消すことはできません。

保存前にテストする

保存前にテスト機能で動作を確認できます。検査文字列（入力・出力フィルタの場合は文章がそのまま渡る、ツール入力・ツール出力フィルタの場合はツールの引数を模した JSON 文字列が渡る）とツール名を指定し、判定スクリプトをその場で 1 回だけ実行します。結果には、動作（止める／記録する／何もしない）、文言、所要時間（ミリ秒）が表示されます。



図：ガードレールのテスト画面。判定スクリプトに対して検査文字列を渡し、動作・文言・所要時間を確認できる。

テスト実行はサーバ側で判定を 1 回動かすだけで、エージェント自体は動きません。想定した入力に対して、意図した動作が返るか、`message` が期待どおり表示されるかを確認してから保存します。

発動状況を確認する

止める（BLOCK）と判定した場合だけが監査ログに記録されます。記録する（LOG）の発動件数は監査ログでは追えないため、トレースのガードレールの検査結果で確認します。

困ったときの確認

状況	確認・対応
止める設定にしたのに何も表示されない	<code>message</code> を返しているか確認する。 <code>message</code> が無いと利用者には何も表示されない
意図せず処理が止まる	判定スクリプトが誤って BLOCK を返していないか、テスト機能で確認する
記録するの件数を監査ログで確認できない	記録する・何もしないは監査ログに残らない仕様。トレースを見る
逐次応答のオーバーラップを保存できない	オーバーラップはウィンドウサイズの半分（小数切り捨て）が上限
複数のガードレールのうち、1件しか発動しない	同一検査位置では登録順に評価され、止めると判定した時点で以降は評価されない
回数を数える判定が意図どおりに動かない	グローバル変数ではなく <code>ctx.state</code> を使っているか確認する。トップレベルの変数は実行1回につき1度しか評価されず、値を実行間で持ち越せない
特定のツールだけを検査する判定が、どのツールにも一致しない（ <code>ctx.tool</code> が空文字になる、または期待した名前にならない）	<code>ctx.tool</code> に値が入るのは、ツール入力・ツール出力の検査位置だけで、それ以外の検査位置では空文字になる。検査位置を確認する。ツール名が重複して連番が付いた場合や、ツール定義側で改名した場合も名前が変わる（「 LLMに公開されるツール名の一意化 」）

精度評価

精度評価は、用意した質問やタスクをエージェントに実行させ、期待する回答や動作と比べる機能です。同じテストケースで評価を繰り返すと、設定の変更によって何が改善し、何が悪化したかを調べられます。

初めて評価する担当者向けに、評価の考え方から結果の確認までを説明します。入力形式や例外条件の詳細は「[設定値](#)」を参照してください。

項目

- 評価で確かめること
 - スコアと業務上の可否
 - 評価種別の選び方
 - 評価に使う情報の単位
- 評価を始める準備
 - 画面を開く
 - 必要な権限
 - 対象とモデルの準備
 - 実行先の確認
- テストケースの作り方
 - 通常の質問と失敗しやすい質問
 - RAGの質問と模範回答
 - エージェントの期待回答と判定条件
 - 自動生成したケースの確認
- 新しい評価計画を作る
 - 「対象と種別」を設定する
 - 「データセット」を準備する
 - 「採点条件」を設定する
 - 「確認」で保存または実行する
- データセットを確認・修正する
- 実行状況とやり直し
- 結果の読み方
 - 件数とエラーを先に確認する
 - RAGの指標から調べる箇所
 - エージェントの指標から調べる箇所
 - スコア平均と前回比較
 - 時間と費用
- 改善して同じ条件で測る
- レポートの保存
- 困ったときの確認
- 設定値
 - 採点条件と対象バージョン
 - データセットの共通制約
 - RAGの入力形式（CSV・TSV・Excel）
 - エージェント評価の入力形式（JSON）
 - 呼び出し条件と証拠
 - 実行のやり直しに必要な条件
 - RAGの対象外条件
 - エージェントの対象外条件
 - 履歴比較の条件
 - 改善提案・権限区分

評価で確かめること

スコアと業務上の可否

スコアは、用意したテストケースと採点条件に対する測定値です。製品が業務上の合格・不合格を自動決定するものではありません。採点できなかったケースや、試していない利用場面まで含めて品質を保証する値でもありません。

例えば、社内規程を検索して回答するエージェントであれば、「規程どおりに答えたか」と「必要な規程を検索できたか」を分けて確認します。検索結果に必要な情報があっても、回答時に金額を取り違える場合があるためです。回答の誤りと検索の取りこぼしでは、見直す設定が異なります。

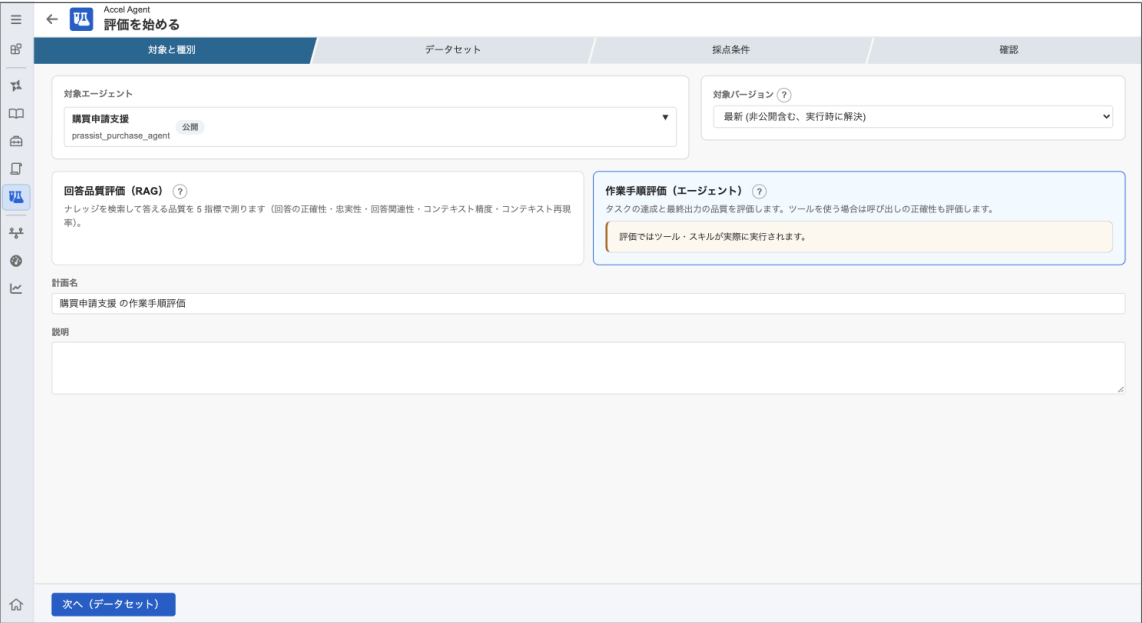
評価を始める前に、利用場面と許容できない失敗を決めます。「回答は読みやすいこと」だけでは担当者によって判断が変わるため、「申請期限と上限額を両方答える」「資料にない金額を作らない」のように、実際の回答から確認できる条件にします。

評価種別の選び方

評価種別	確かめる内容	用意する主な期待値
回答品質評価（RAG）	ナレッジを検索して回答する品質	質問と模範回答
作業手順評価（エージェント）	呼び出しの適切さ、タスクの達成、最終回答の品質	タスク指示、期待する呼び出し、期待回答（画面では「期待する回答（模範回答）」）または判定条件

RAG（検索拡張生成）は、文書などのナレッジを検索し、その情報を使って回答する方法です。RAG評価には、対象バージョンにナレッジの設定が必要です。RAG評価の採点では、回答の正確性を含む5指標を使います。

エージェント評価では、ツールを使うタスクに加え、文章の要約・変換なども判定条件で表現できます。どの指標が採点されるかは、評価種別を選んだだけでは決まらず、テストケースに設定した期待値によって変わります。例えば、期待する呼び出しも禁止ツールも設定しなければ、ツール正確性は採点されません。



図：「評価を始める」の最初の画面。対象エージェント・対象バージョンと、評価種別（回答品質評価／作業手順評価）を選ぶ。

評価に使う情報の単位

用語	意味
評価計画	評価対象、評価種別、データセット、既定の採点条件をまとめた設定
データセット	1～100件のテストケースのまとまり
テストケース	1回分の質問またはタスクと、その入力・期待値
評価実行	計画に基づいて測定した1回分の記録。計画内に実行番号が付く
採点モデル	回答や実行記録を読んで判定するLLM（大規模言語モデル）
Embeddingsモデル	文章を数値の列へ変換し、文章どうしの類似度を計算するためのモデル
採点回数	同じ回答に対して、一部の指標の判定を繰り返す回数

通常、対象エージェントはテストケースごとに新しいセッションで1回実行されます。採点回数を3にすると、対象エージェントを3回動かすのではなく、保存した同じ回答を繰り返し採点します。採点回数を増やすと、採点のばらつきが抑えられ、スコアが安定します。対象エージェントの回答のばらつきを調べるには、評価実行そのものを繰り返します。

評価を始める準備

画面を開く

サイトマップから「Copilot」→「Accel Agent」→「精度評価」を開きます。Accel Agent 内では、サイドナビゲーションの「精度評価」から移動できます。

精度評価ホームでは、「自分の評価計画」「最近更新した評価計画」「最近の評価結果」を確認できます。新しく作成する場合は「評価を始める」、既存の評価を続ける場合は計画を開きます。

必要な権限

操作には評価の権限が必要です。参照権限で計画や結果を確認し、管理権限でデータセット画面の表示、計画の作成・変更、実行、生成などを行います。改善提案と添付ファイルの参照にも管理権限が必要です。利用者に見える操作は、付与された権限によって異なります。権限の詳細は「[参照だけでは足りない場面](#)」を参照してください。

対象とモデルの準備

評価には、対象エージェント、採点用のLLMモデル、Embeddingsモデルが必要です。自動生成する場合は、テストケースを生成するLLMも選択します。これらのモデルは、「[AI モデル管理](#)」に登録したモデルから選ぶため、採点用・生成用にはテキスト生成モデルを、Embeddingsモデルには埋め込みモデルを、あらかじめ登録しておきます。生成用・採点用のモデルを選んでも、対象エージェント自身が使うモデルは変更されません。

実行先の確認

評価では、エージェントに設定したツールやスキルが実際に動作します。登録・更新・通知を行うツールを使う場合は、評価で変更してよい接続先とデータを用意します。これはRAG評価でも同じで、評価種別を選ぶだけで外部操作が無効になる仕組みはありません。

禁止ツールの指定は、呼び出しを検出して減点する採点条件です。実行を事前に遮断する設定ではないため、外部操作の防止を禁止ツール欄だけに任せることはできません。

テストケースの作り方

通常の質問と失敗しやすい質問

データセットには、普段の利用を代表する質問と、失敗すると困る質問を含めます。通常の質問だけでは、情報が見つからない場合や、誤った操作を求められた場合の振る舞いを確認できません。

種別	種類タグ	確認する状況
RAG	正常 (NORMAL)	資料に答えがある
RAG	回答が見つからない (NOT_FOUND)	資料に答えがなく、その旨を答える
RAG	曖昧 (AMBIGUOUS)	質問の条件が曖昧で、適切な扱いが必要
エージェント	正常 (NORMAL)	通常の指示を実行する
エージェント	ツール誤用誘発 (TOOL_MISUSE)	不適切な呼び出しを誘う指示に対応する
エージェント	ガードレール (GUARDRAIL)	エージェント定義のガードレールが、期待どおり通過・記録・遮断する
エージェント	未完遂 (UNFULFILLABLE)	実行できない依頼を適切に断る、または不足条件を説明する

種類タグはケースの意図を表します。例えば、ガードレールのタグを付けても禁止ツール一覧が自動で補われるわけではありません。何をよく、何をしてはいけないかは期待値に記入します。指示文の制約によって断らせたいケースは、ガードレールではなく未完遂 (UNFULFILLABLE) として作成します。自動生成でも、ガードレールを設定していないエージェントでは、ガードレールに割り当てた件数を未完遂として生成します。

対象エージェントにガードレールを設定している場合は、種類タグとは別に、テストケースごとに「ガードレール評価」を指定できます。期待する動作（通過・記録のみ・遮断）と、必要に応じて対象ルールと検査位置（入力・ツール実行前・ツール実行後・出力）を指定すると、実行時に記録した実際の判定と照合して「適合」「不適合」を判定します。照合にLLMは使いません。判定処理の失敗によって通過した場合は、適合とは判定しません。指定は、データセットの管理の行編集、またはインポートするファイルのguardrail項目（「[RAGの入力形式 \(CSV・TSV・Excel\)](#)」、「[エージェント評価の入力形式 \(JSON\)](#)」）で行います。

遮断を期待したケースでは、期待どおり遮断されれば回答を採点しないため、期待回答を空にできます。通過・記録のみを期待したケースは、通常のケースと同様に回答やタスク完遂も採点します。

RAGの質問と模範回答

模範回答には、業務資料で確認した答えを記入します。次は説明用の架空の規程を使った例です。

項目	例
規程の記載	購入補助の上限は10万円。申請は購入月の月末まで
質問	購入補助の上限額と申請期限を教えてください
模範回答	上限は10万円です。購入月の月末までに申請してください

質問で二つの事項を求めるなら、模範回答にも両方を含めます。模範回答から期限が抜けていると、採点でも期限の欠落を十分に評価できないためです。規程が改定されると模範回答も変わるため、模範回答の根拠にした資料の版や適用日を、評価計画の説明欄に書いておきます。

NOT_FOUNDでは、質問への具体的な答えを捏造せず、情報がないことを伝える回答を期待値にします。この種類は回答の正確性で採点し、忠実性などの残り4指標は対象外です。

エージェントの期待回答と判定条件

固定の模範回答を用意できる場合は、期待回答を基準にする方式 (REFERENCE) を使います。決まった文書の要約や、既知の内容への回答が該当します。

実行時に取得する値が変わる場合は、条件方式 (CRITERIA) を使います。例えば在庫照会では、事前に架空の在庫数を正解として固定するより、「照会結果と同じ在庫数を回答する」と定める方が、測りたい振る舞いに合います。

条件方式では、タスク完遂の条件と回答品質の条件を分けます。タスク完遂は必須条件をすべて満たしたときに1点、回答品質は満たした条件の割合を基本に採点します。回答中の事実根拠があるかも確認し、根拠のない事実があると回答品質は0点です。

条件の例	使用する証拠	判断できること
回答を3項目の箇条書きにする	ANSWER	指示・入力と最終回答の関係
指定された商品コードで照会する	TOOL_CALLS	呼び出し先・引数・エラーの有無
照会結果と同じ在庫数を答える	TOOL_RESULTS	呼び出し記録と実際の応答の関係

「処理が完了した」と回答に書かれているだけでは、外部処理が完了した証拠として扱えません。業務処理の結果を判定する条件には、TOOL_RESULTSなど、完了を確認できる証拠を選びます。ツールが受付までしか返さない場合は、その応答から処理完了までは確認できません。

各条件群は1～10件で、各説明は2,000文字までです。同じ説明の重複は登録できません。条件は、達成したかどうかを一つずつ判断できる単位に分けます。

条件方式のタスク完遂は、条件ごとに指定した証拠（ANSWER・TOOL_CALLS・TOOL_RESULTS）で判定するため、最終回答が空でも、ツールの呼び出しや応答を証拠にした条件であれば0点ではありません。文章による報告を求めない処理を条件方式で評価できるのはこのためです。空回答を理由に0点になるのは、参照回答方式のタスク完遂と、両方式共通の最終出力の品質だけです。

必要なツール応答が欠けている場合や、証拠として渡すツール応答が一定の長さを超える場合は、その条件が判定不能です。大きなツール応答を扱う処理を評価するとき、この制約を踏まえて結果を読みます。

自動生成したケースの確認

自動生成は、テストケースのたたき台を作る機能です。生成後に担当者が質問、模範回答、呼び出し先、判定条件を確認します。生成処理の検査を通ったことと、業務上の正解であることは別です。

RAGの自動生成では、現在の検索で根拠を取得できるかも検査します。そのため、資料に答えはあるのに検索で見つけられないケースが生成結果から除かれることがあります。検索の取りこぼしを評価する場合は、そのようなケースを人手で補います。

比較に使うデータセットは、確認後に固定します。毎回生成し直すと、対象の改善に加えて質問の難しさも変わり、スコア差の原因を切り分けられません。

新しい評価計画を作る

「対象と種別」を設定する

「評価を始める」をクリックし、対象エージェント、対象バージョン、評価種別、計画名を設定します。説明欄には、評価の目的と、比較する変更内容を記入しておくとかから確認できます。

対象バージョンの「最新」は、実行開始時に非公開を含む最新の番号を選びます。公開中のバージョンだけを意味するものではありません。番号を固定して測定する場合は、そのバージョンを明示的に選択します。

対象バージョンは計画の固定設定として保存されません。「保存のみ（実行しない）」で計画を作った後や、次の評価を始めるときも、実行前に指定を確認します。実際に使用したバージョンは結果の「実行条件」で確認できます。

「データセット」を準備する

データセットは、自動生成、インポート、既存計画からの取り込みのいずれかで準備します。

方法	操作	確認する内容
自動生成	名前、件数、種類タグの配分、追加指示、生成モデル、Embeddingsモデルを設定し、生成を開始する	生成件数、警告、質問と期待値
インポート	RAGは記入用テンプレートを取得し、エージェント評価は作成用コンテキストを確認したうえで、作成したファイルを選択する	読込件数、種類タグ、行ごとの警告
既存計画から	同じ評価種別の計画を選ぶ	対象エージェントでも入力や呼び出し先が有効か

RAGの取り込み形式はCSV、TSV、Excel（.xlsx）です。CSVとTSVはUTF-8で保存します。エージェント評価はJSON形式です。ファイルの項目名と記入例は「[設定値](#)」に掲載しています。

エージェント評価には、RAGのような固定の記入用テンプレートがありません。呼び出せるツール・ナレッジベースや入出力スキーマが対象エージェントごとに異なるためです。代わりに、インポート画面の「作成用コンテキストを表示」から、対象エージェントに実在する呼び出し候補（種別・識別名・入力スキーマ）と入力パラメータのスキーマ、そのまま書き足せる取込ファイルの骨組みを確認できます。ここに表示される呼び出し候補とスキーマは、同じ画面の「自動生成」が生成時に参照する材料と共通です。「すべてコピー」でコピーした内容は、人が直接埋めるほか、手元のAIアシスタントに渡して記入内容を一緒に考える使い方もできます。



図：データセットのインポート画面。JSON ファイルをドロップ、またはファイルを選択して取り込む。

自動生成の件数は1~100件、初期値は30件です。配分は画面で合計100%にします。「数件だけ試す」は生成内容を確認するためのプレビューで、対象エージェントの少数実行とは別の操作です。通常の自動生成は計画とデータセットも作成するため、完了後の内容を確認してから次へ進みます。

「採点条件」を設定する

採点モデル、Embeddingsモデル、採点回数を選択します。

採点回数の初期値は3回です。忠実性、回答関連性、回答の正確性、タスク完遂率、最終出力の品質で反復判定を行います。コンテキスト精度とコンテキスト再現率は、採点回数を増やしても各1回の採点です。

条件をそろえて比較する場合は、採点モデルだけでなく、Embeddingsモデルと採点回数も固定します。Embeddingsモデルを変更すると、文章の類似度や引数照合の結果が変わる場合があります。

「確認」で保存または実行する

設定とデータセットを確認し、「保存のみ（実行しない）」または「実行して結果へ」を選択します。初回は先頭数件だけを実行し、入力、接続先、所要時間、採点理由を確認すると、全件実行前に設定の問題を見つけられます。

少数実行は、指定した件数を処理した後に中断状態へ移ります。残りを続ける場合は再開します。少数実行の平均は全件の平均とは扱われず、全件評価の比較対象には入りません。

データセットを確認・修正する

計画から「データセットの管理」を開くと、ケースの入力と期待値を確認できます。テストケースを編集した後は保存し、検証警告を読みます。「問題なし」は設定上の検査結果であり、人手による正解確認済みという意味ではありません。

同じ計画へのインポートでは、ケースの件数と、種類タグごとの件数を変えられません。件数や種類の構成が変わるとスコアの母数と意味合いが変わり、同じ計画の中で前回と比べられなくなるためです。既存ケースの内容を直す場合は、エクスポートした原本を修正してインポートし直します。ケースを増やしたり減らしたり、種類タグの配分を変えたりする場合は、新しい評価計画を作ります（「[新しい評価計画を作る](#)」）。

画面の行編集は既存ケースを直す補助手段であり、行の追加・削除はできません。ケースの質問・入力・模範回答などを変えると、変更前の実行とは比較条件がそろわなくなり、前回比が表示されなくなります（「[履歴比較の条件](#)」）。どの変更がスコアに影響したかを後から確かめたり、元に戻したりできるよう、修正前にエクスポートしたファイルを保管し、変更した内容を評価計画の説明欄に書いておきます。

実行ユーザを指定する場合は、そのユーザの権限で確認したいケースであることを確かめます。未指定の場合は、評価を開始したユーザとして実行されます。権限によって見えるナレッジや実行できる処理が変わるため、比較時にはこの指定もそろえます。

構造化出力の一部分を評価する場合は、ケース編集の「評価対象のプロパティ」を選択します。例えば出力が `{"summary":"要約文","count":3}` なら、summaryだけを回答品質の評価対象にできます。配列の要素番号は0から始まります。「回答全体」を選ぶと指定を解除できます。

テストケースには画像を添付できます。ケース編集の「画像添付」で「画像を追加」から登録すると、エージェント評価では、実行時にその画像が対象エージェントへ渡されます。画像を読み取って処理するエージェントを評価する場合に使います。添付できる形式はPNG・JPEG・GIF・WebPで、1ファイルの上限は5MiBです。インポートするファイルからは画像を登録できないため、画像はデータセットの管理から登録します。

プロパティを指定したRAG評価では、回答関連性を算出しません。エージェント評価では、最終出力の品質だけに指定が適用され、タスク完遂と呼び出し判定は実行全体を見ます。定義変更で選択したプロパティがなくなった場合は、選び直すか解除します。

RUNNINGまたはSTOPPINGの評価がある間は、同じ計画のデータセット変更が制限されます。中断中はデータセットを変更できますが、評価の開始後にケースを変更すると、その評価は再開できなくなります。中断した評価の残りを処理したい場合は、再開して実行が終わるまでデータセットを変えずにおきます。

実行状況とやり直し

状態	意味	次の確認・操作
実行中（RUNNING）	ケースを実行・採点している	処理件数を確認する。必要なら中断を要求する
停止処理中（STOPPING）	中断要求を受け付け、処理中のケースの終了を待っている	中断状態になるまで待つ
中断中（SUSPEND）	中断要求または少数実行の上限で止まった	未処理のケースを再開できる
完了（OK）	実行の処理を終えた	ケースのエラー件数と各指標を確認する
エラー（ERROR）	実行全体の処理を継続できなかった	エラー理由を確認する

「完了」は全テストケースの回答が正しかったという意味ではありません。低スコアやテストケース単位のエラーがあっても、全件を最後まで処理すれば「完了」と表示されます。「完了」は評価の実行が最後まで走りきったことだけを表します。

やり直す操作は、対象エージェントを動かすかどうかで選びます。

操作	対象エージェント	使う材料	用途
もう一度実行	実行する	現在のデータセットと指定した対象バージョン	変更後の品質を測る
再開	未処理ケースを実行する	同じ実行のデータセット内の未処理ケース	中断・少数実行の続きを行う
この行を再実行	該当ケースを実行する	現在の該当ケース	対象実行の失敗をやり直す
保存結果を再採点	実行しない	保存済みの入力・期待値・回答・呼び出し記録	保存した回答の判定をやり直す

「再開」は、既にエラーとして記録したケースを自動で再実行しません。エラーケースは、結果画面の「この行を再実行」で個別に再実行します。「保存結果を再採点」は同じ実行の採点結果を更新し、その実行に保存された採点条件を使います。新しい対象設定を試す場合は、計画画面の「もう一度実行」で新しい評価を実行します。

結果の読み方

件数とエラーを先に確認する

結果では、総件数、処理済み件数、エラー件数、指標ごとの「採点値あり」の件数を確認します。例えば20件中2件にしか採点値がない0.9と、20件すべてを採点した0.9では、結果から判断できる範囲が異なります。



図：評価結果の画面。指標ごとのスコア、ガードレール評価の件数、評価件数（種類ごとの内訳）、テストケース一覧が並ぶ。

表示	読み方
0点	採点条件に対して0と判定した、または空回答などを0と定義している
対象外・「ー」	模範回答がない、指定プロパティで対象外になるなど、値を算出していない
判定不能	条件判定に必要な証拠が不足している。対象外理由を確認する
エラー	対象実行または採点処理が失敗した。回答品質の0点と区別する

対象外とエラーは、欠けた指標を0として平均へ加える扱いではありません。平均が高くて、採点されなかったケースに問題が残る場合があります。

ガードレール評価を指定したケースがある場合は、結果画面の上部に、適合・不適合・評価不能などの状態ごとの件数が表示されます。評価不能は、対象ルールが実行時の定義にない場合や、前の段階で遮断されて対象の検査まで到達しなかった場合などです。平均値とあわせて、この件数も確認します。

RAGの指標から調べる箇所

Accel Agent 利用ガイド — Accel Agent 利用ガイド 初版 2026-10-01

指標	測定内容	低い場合に確認する箇所	主に見直す設定
回答の正確性	模範回答に合う内容を答えたか	金額・期限・条件の誤り、回答の欠落	指示（「 指示を書く 」）。検索結果に必要な情報がなければ、コンテキスト再現率と同じ箇所
忠実性	回答中の主張を検索根拠で裏付けられるか	文書にない事実や条件の付け足し	指示で、資料にない事実を補わないよう制約する（「 指示を書く 」）
回答関連性	回答から逆に作った質問が元の質問に近いか	質問に答えているか、話題がずれていないか	指示で、回答の範囲や形式を定める（「 指示を書く 」）
コンテキスト精度	取得した情報のうち模範回答に役立つ割合	無関係な検索結果が多くないか	Top-K・閾値・メタデータフィルタ（「 ナレッジベース 」）、チャンク分割（「 チャンク分割を設定する 」）
コンテキスト再現率	模範回答に必要な情報を取得できた割合	必要な文書・段落の取りこぼし	ナレッジの登録内容（「 アイテムを取り込む 」）、チャンク分割（「 チャンク分割を設定する 」）、Top-K・閾値

先の補助制度の例で、検索結果には上限と期限の両方があるのに回答が期限を誤ったなら、回答生成を調べます。検索結果に期限の情報がなければ、ナレッジの登録内容や検索を調べます。指標だけで原因を断定せず、ケース詳細の実際の回答と検索根拠を突き合わせます。

エージェントの指標から調べる箇所

指標	測定内容	低い場合に確認する箇所	主に見直す設定
ツール正確性	期待する呼び出し・引数・必要な順序と の一致	呼び出し漏れ、引数の違い、禁止ツールの 呼び出し	ツールの「指示」と入力パラメータの説明 （「 ツール 」）、エージェントの指示での呼び 出し順の指定（「 指示を書く 」）
タスク完遂率	期待する目的を達成したか	実際の処理結果、必須条件の未達	指示、ストラテジと最大ループ回数（「 実行制 御を設定する 」）、必要なツールの割り当て
最終出力の品質	必要な内容を満たし、根拠に沿って回答 したか	必須情報の欠落、結果の誤読、架空の成 功報告	指示での出力内容の指定、構造化出力の定義 （「 入出力値を設計する 」）

ツールの正確性が1でも、ツールの業務処理が成功したとは限りません。この指標は呼び出し先と引数などを照合するため、業務処理の成否はタスク完遂の理由と実際のツール応答で確認します。

禁止ツールを呼ぶと、ツール正確性は0点になり、結果に「禁止ツールを実行しました」と表示されます。この表示はスコアとは無関係に常に出るため、平均スコアとは分けて確認します。

Account Agent

評価結果

スコア詳細 | 機能: すべて ▾

▼ [06] 見積書の内容を確認して支援結果を作成してください。 0.16 未完了

タスク指示

見積書の内容を確認して支援結果を作成してください。

⚠️安全

呼び出しの比較

期待 0 件、一致 0 件、実数 2 件
 期待する呼び出しは指定されていません。

⚠️安全 禁止ツールを実行しました prassist_vendor_lookup_flow
 実際の呼び出し: TOOL prassist_vendor_lookup_flow → KNOWLEDGE 85k3b53osrtn1

期待する回答 (模範回答)

品目が読み取れないため linetems - accountCandidates を空にし、品目なしをwarningsに記す。見積書を讀んだと誤って品目を捏造しない。取引先照会ツールは呼ばない。

実際の回答

```
[{"vendorName":"連携テクノロジー株式会社","totalAmount":0,"linetems":{"accountCandidates":{"businessReason":"見積書に実施理由・背景の記載はありません。申請者は以下の観点を具体的に記入してください：\n- 新規導入か買い替えか増員か\n- 導入の目的（例：システム更新、開発環境構築、消耗品補充、顧客向け納品など）\n- 期待する効果・成果\n- 必要期間および必要数量\n- 予算の概算や関連プロジェクト\nこれらを明確にすることで審査が容易になります。"},"accountReason":"見積書に品目情報がないため、勘定科目マスタとの対応を特定できません。品目名と用途（消耗品、fixed asset、ソフトウェア、外注費等）を提示いただければ、勘定科目マスタを参照して該当する科目候補を提示します。表にない品目については推測で科目コードを作成しません。"},"vendorReason":"取引先照会の結果、連携テクノロジー株式会社は取引先マスタに登録済（取引先コード：DEMO-V-001）でした。既存取引先であることから、過去の取引履歴・保守対応の継続性・支払条件の確認がしやすい点が購入先決定理由として挙げられます。具体的な選定根拠（価格・納期・技術サポート等）は申請者が明記してください"}]
```

図：テストケース詳細。呼び出しの比較（期待する呼び出しと実際の呼び出し）、期待する回答と実際の回答を確認できる。

スコア平均と前回比較

スコア平均は、算出できた品質指標をまとめた診断用の値です。RAGでは回答の正確性が平均の上限です。例えば他の4指標がすべて1でも、回答の正確性が0ならスコア平均も0です。時間・コスト・推論回数は、この平均に含めません。

前回比と過去最高比は、同じ計画で比較条件がそろった全完了結果を対象とします。題材、入力、期待値、実行ユーザ、採点条件、指標ごとの採点件数などが異なる場合は、比較値が表示されないことがあります。対象エージェントのバージョン変更は改善の比較に必要なため、それだけで比較対象外にはなりません。

前回は「今回のスコア - 比較対象のスコア」です。0.60 から 0.70 であれば、差は +0.10 と表示されます。これはスコアの差であり、比率ではありません。また、指標値はテストケースごとのスコア (0~1) の平均であり、必ずしも正解したテストケースの割合ではないため、「正解率が 10% 改善した」という読み方はできません。品質指標は高い方を、時間・コスト・推論回数は低い方を良い値として表示しますが、処理を省略して速くないか回答と達成条件でも確認します。

時間と費用

Accel Agent 利用ガイド — Accel Agent 利用ガイド 初版 2026-10-01

エージェント評価には、対象側の実行時間、コスト、推論回数も表示されます。時間と推論回数はケース間の平均、対象コストは算出できたケースの合計です。コストの有効件数が足りない場合は、その合計だけで全件分の費用を判断しません。

採点・埋め込みの費用と、対象エージェントの費用は別に記録されます。自動生成や改善提案でもLLMを使うため、評価画面の一つの金額を、これらをすべて含む請求額と解釈しないようにします。単価や使用量を取得できない場合の「―」は、無料という意味ではありません。

改善して同じ条件で測る

結果のケース詳細で、期待値、実回答、判定理由、検索・呼び出し記録を確認します。LLMによる判定にも誤りがあるため、判定理由が実際の証拠と合っているかを担当者が確かめます。

改善提案は、評価結果からLLMが作る修正案です。対象の定義を自動変更する機能ではありません。提案の根拠となるケースを読み、変更が妥当かを判断してからエージェントやナレッジを修正します。保存済みの提案を閲覧するだけでは再生成されません。

改善後は、固定した同じデータセットと採点条件で新しい評価を実行します。複数の設定を同時に変えると原因を分けにくくなるため、何を変更したかを評価メモに記録します。判定モデルを変えた場合は、対象の改善と採点基準の変化を区別して扱います。

調整に使った質問だけで判断すると、その質問にだけ合う修正を採用するおそれがあります。最終確認には、調整に使わなかった別のデータセットも用意します。人手での判定と比べ、「誤答を正しいと判定したケース」と「正答を誤りと判定したケース」を分けて数えると、採点モデルの癖も確認できます。

レポートの保存

結果画面からHTMLレポートを取得すると、指標、採点件数、ケース詳細などをまとめて確認できます。改善提案の掲載は取得者の管理権限に従います。

詳細データには保持期間を設定できるため、長期保管する結果は、必要なケース詳細を参照できるうちにレポートとして保存します。保持期間の既定値は「精度評価メンテナンス」ジョブの180日ですが、出荷時には自動実行のスケジュールがありません。実際の削除時期は環境の運用設定によります。保持期間の変更やジョブの設定は「[ジョブごとの設定](#)」を参照してください。

困ったときの確認

評価の準備や実行がうまく進まない場合、または結果が想定どおりに表示されない場合は、次の表で状況に合う項目を確認します。

状況	確認・対応
RAGを選択できない	選んだ対象バージョンにナレッジがあるか確認する
自動生成の件数が指定より少ない	生成の警告と採用件数を確認し、必要なケースを手手で補う
モデルのエラーが出る	採点モデルとEmbeddingsモデルの選択と、採点モデルが構造化出力に対応しているかを確認する
指標が「―」になる	ケースの模範回答、検索記録、評価対象プロパティ、対象外理由を確認する
条件判定で証拠が不足する	ツール応答の欠落・切り詰めと、選んだ証拠の種類を確認する
停止処理中が続く	実行中のケースが終了しているか確認する。中断要求は即時強制終了ではない
再開・再実行で定義変更が検出される	同じ実行への継ぎ足しを避け、変更後のバージョンで新しい評価を開始する
前回比が出ない	全件完了、エラー件数、題材と採点条件、有効件数が一致しているか確認する
再採点できない	保存済みの回答とケースのスナップショットが残っているか確認する
「作作用コンテキスト」の「すべてコピー」をクリックしても反応が無い	HTTPでアクセスしていないか確認する。ブラウザのクリップボード機能はHTTPS（または http://localhost ）でしか動作せず、素のHTTPではクリックしても何も起きない。ダイアログに全文が表示されているので、範囲選択して手動でコピーする

設定値

本節では、ファイルを作成するときに必要な項目名・制約と、結果画面に表示される状態の意味をまとめます。スコアが上下する理由は、「[結果の読み方](#)」の指標の説明を参照してください。

採点条件と対象バージョン

実行開始時に指定した値を優先し、未指定の項目は計画の既定値で補います。

項目	意味・制約
採点モデル	採点に使用するモデル。AIモデル管理に登録したテキスト生成モデルから選ぶ
Embeddingsモデル	AIモデル管理に登録した埋め込みモデルから選ぶ
採点回数	反復する指標の採点回数（正の整数）。既定値は3

採点モデルが構造化出力を返せない場合、対象エージェントを1件も実行しないまま実行全体をエラーにします。

対象バージョンは実行時に指定し、省略時は非公開のバージョンを含めて取得できる最大番号を選びます。計画自体は対象バージョンを固定保持しません。受付時に対象バージョンと定義の内容から計算した識別値（ハッシュ値）を保存し、対象実行の直前に定義を取得し直して比較します。不一致であれば、その対象を実行せずにエ

ラーにします。実行の途中で定義を変更しても、その回の実行には反映されません。

「再開」「この行を再実行」でも、この識別値を確認します。「保存結果を再採点」は保存済みの回答を使うため、定義が変わっていても対象を再実行せずに処理できます。また、ナレッジ本文や外部業務データを実行時点へ完全に復元する仕組みはないため、同じ定義であっても、これらの変化によって結果が変わることがあります。

データセットの共通制約

データセットは1～100件です。ケースID は最大50文字で、同じデータセット内の重複は拒否されます。未指定のID は自動採番します。質問またはタスク指示は必須です。

検証状態の「未検証」「問題なし」「警告あり」は、項目や定義との整合性を確認した結果です。模範回答の事実確認や、採点モデルの判定精度を保証する状態ではありません。

RAGの入力形式（CSV・TSV・Excel）

CSVとTSVはUTF-8で読み込み、先頭のBOM（バイトオーダーマーク）を許容します。Excelは.xlsxの先頭シートを読み、数式セルは数式の評価結果を使います。先頭行はヘッダとして扱います。question列とground_truth列は必須ですが、ground_truthの値そのものは空欄でも構いません。

列名	内容	省略・空欄時の扱い
id	ケースID	自動採番
kind	NORMAL、NOT_FOUND、AMBIGUOUS	未指定はNORMAL。不明な値は警告した上でNORMALとして扱う
question	質問	必須
ground_truth	模範回答	空欄は警告。模範回答を必要とする指標は対象外になる
target_path	構造化出力のうち評価対象とするプロパティ	省略時は回答全体
run_user	実行ユーザコード	省略時は評価を開始したユーザとして実行される
param:入力名	エージェントの入力パラメータ	入力定義との対応を検証する
guardrail	ガードレール評価の期待条件。JSON文字列でaction・guardrailId・phaseを指定する（各項目の意味は「 エージェント評価の入力形式 (JSON) 」と同じ）	省略時はガードレール評価の対象外

記入例（説明用の架空の規程を使ったもの。実際の評価では対象資料に合わせて差し替えます）。

```
id,kind,question,ground_truth,target_path,run_user
rag-01,NORMAL,購入補助の上限額と申請期限は？,上限は10万円で、申請期限は購入月の月末です。,,
rag-02,NOT_FOUND,翌年度の購入補助の上限額は？,提供された資料には翌年度の上限額の記載がありません。,,
```

必須ヘッダや質問が欠落している場合や、ケース数が上限を超える場合は、取り込みエラーが発生します。行末の空セルのように、存在しない位置の値は空として扱います。取り込み後は、追加の入力パラメータや種類タグの警告を確認してから実行します。

エージェント評価の入力形式（JSON）

ルートはオブジェクトとし、文字列のformatVersion: "1.0"と、配列のsamplesを指定します。JSON内のキー重複は拒否されます。未知の項目は警告した上で読み飛ばすため、期待条件の入力ミスが警告のまま見過ごされないよう、取り込み後の警告を確認します。

項目	型・内容
id	文字列。未指定は自動採番
kind	NORMAL、TOOL_MISUSE、GUARDRAIL、UNFULFILLABLE
taskInstruction	必須のタスク指示
expectedAnswer	参照回答方式（REFERENCE）で使う期待最終回答
inputParameters	入力名と値のオブジェクト。値はスカラー値を文字列として扱う。配列・オブジェクト・nullは指定できない
runUserCd	任意の実行ユーザコード
expectedCalls	期待する呼び出しの配列（「 呼び出し条件と証拠 」）
forbiddenTools	禁止するツール名の文字列配列
answerEvaluationMode	REFERENCEまたはCRITERIA。未指定はREFERENCE相当
completionCriteria	条件方式（CRITERIA）のタスク完遂条件の配列
answerCriteria	条件方式（CRITERIA）の回答品質条件の配列
callOrderRequired	条件方式で呼び出し順を要求するかどうか（真偽値）

項目	型・内容
targetPath	最終出力のうち品質を採点するプロパティ。画面の「評価対象のプロパティ」で選ぶ値に当たる。単一プロパティまたは配列要素を指定でき、記法の例は\$.summary、\$["日本語のキー"]、\$["items"][0]["name"]、\$[0]。ワイルドカード・フィルタ・再帰検索には対応しない
guardrail	ガードレール評価の期待条件（オブジェクト）。action（必須。PASS・LOG・BLOCK）、guardrailId（対象ルールのID。省略時はすべてのルール）、phase（INPUT・TOOL_CALL・TOOL_RESULT・OUTPUT。省略時はすべての位置）を指定する。対象エージェントの定義にないルールを指定するとエラーになる。actionがBLOCKの場合はexpectedAnswerを空にできる。kindのGUARDRAILとは別の項目
attachments	添付画像の情報（名前・形式など）。エクスポートしたファイルに含まれるが、画像そのものは含まれず、インポートしても画像は登録されない。画像はデータセットの管理から登録する

記入例（入力文の変換を条件方式で評価する例。ツールを必要としないため呼び出し期待値は指定していません。取り込むだけでは対象エージェントは実行されません）。

```
{
  "formatVersion": "1.0",
  "samples": [
    {
      "id": "agent-01",
      "kind": "NORMAL",
      "taskInstruction": "次の文を丁寧な日本語に直してください。『資料を明日までに送って。』",
      "answerEvaluationMode": "CRITERIA",
      "completionCriteria": [
        { "description": "資料の送付を依頼する意味と、明日までという期限を保持する", "evidence": "ANSWER" }
      ],
      "answerCriteria": [
        { "description": "丁寧な日本語の依頼文になっている", "evidence": "ANSWER" },
        { "description": "変換後の文だけを出力する", "evidence": "ANSWER" }
      ],
      "callOrderRequired": false
    }
  ]
}
```

呼び出し条件と証拠

期待する呼び出しにはtype・name・任意のargsを指定します。typeはTOOL・KNOWLEDGE・SKILLのいずれかです。nameには実際の呼び出しカタログの識別名を使います。表示名を推測して記入すると一致しないことがあるため、対象の候補を評価画面側で確認してから記入します。

argsの各項目はname・expected・matchを持ちます。matchはEXACT（前後空白を除いて比較し、双方が数値なら数値として比較する。1と1.0は一致）、またはSIMILARITY（期待引数と実引数の埋め込みのコサイン類似度が0.85以上で一致）のいずれかです。

```
{
  "type": "TOOL",
  "name": "lookup_inventory",
  "args": [
    { "name": "productCode", "expected": "P-001", "match": "EXACT" }
  ]
}
```

条件方式（CRITERIA）のcompletionCriteria・answerCriteriaは、それぞれ1～10件を指定します。各条件はdescriptionとevidenceを持ち、descriptionは空白だけを認めず最大2,000文字です。同じ条件群の中で、前後の空白を除いた説明が完全一致する条件は拒否されます。参照回答方式（REFERENCE）では条件群を指定できません。

evidenceに指定できる値は、「エージェントの期待回答と判定条件」で挙げたANSWER・TOOL_CALLS・TOOL_RESULTSのいずれかです。TOOL_RESULTSでは、全呼び出しの応答が完全に保存されていること（resultCompleteが真であること）を要求します。応答が未保存・一部欠落の場合や、呼び出し要素自体が欠けている場合は証拠不足として扱われます。TOOL_CALLSとTOOL_RESULTSの材料は、JSON化した合計が24,000文字を超える場合、証拠不足として扱われます。呼び出しが0件であることは、記録の欠落ではなく「呼び出さなかった」という観測として扱います。

ツールの応答本文は1件あたり16,000文字を上限に保存します。条件判定に必要な応答が切り詰められた場合は、切り詰めた値だけで条件を満たしたとは判定せず、その指標は算出せず「判定不能」として扱います。

実行のやり直しに必要な条件

同じ計画で実行中（RUNNING）または停止処理中（STOPPING）の実行がある間は、新しい実行を受け付けません。

対象実行でrunUserCdを指定すると、その間だけ指定ユーザとして実行します。終了後に元のユーザへ戻す処理に失敗すると、誤ったユーザのまま処理を続けないよう実行全体を停止します。

操作	開始条件	処理内容
再開	中断中（SUSPEND）、モデル利用可能、定義ハッシュ一致、開始後にデータセットが変更されていない	既存結果のないケースだけを処理する。記録済みのエラーケースは自動で再実行しない

操作	開始条件	処理内容
この行を再実行	実行中・停止処理中ではなく、対象ケースがエラー	現在のデータセットの指定ケースを対象バージョンで再実行する
保存結果を再採点	実行中・停止処理中ではなく、必要な保存明細がある	保存済みのケーススナップショット・回答・根拠・呼び出しで判定をやり直す

再採点で、明細・ケーススナップショット・回答のいずれかが欠けているケースは、既存値を保持してスキップします。再開は、評価の開始後にデータセットが変更されていると拒否されます。「この行を再実行」は現在のケース内容を読むため、停止中に編集すると、一つの実行内に異なる時点のケース内容が混在することがあります。内容を変更して比較したい場合は、新しい評価実行を作成します。

RAGの対象外条件

次の表は、NOT_FOUND以外のケースを対象とした、指標ごとの対象外条件です。「通常」は他の条件を満たした場合に通常どおり採点する、という意味です。

条件（画面の対象外理由）	正確性	忠実性	回答関連性	コンテキスト精度	コンテキスト再現率
検索記録なし（「検索が行われませんでした」）	通常	対象外	通常	対象外	対象外
検索したที่本文0件（「検索結果が 0 件でした」）	通常	回答があれば0	通常	対象外	模範回答があれば0
模範回答なし（「期待する回答（模範回答）が未設定です」）	回答があれば対象外	通常	通常	対象外	対象外
回答なし（「回答がありませんでした」）	0	対象外	0	通常	通常
targetPath指定	抽出値を採点	抽出値を採点	対象外	通常	通常
targetPath抽出失敗	0	対象外	対象外	通常	通常

NOT_FOUNDは回答の正確性だけを採点し、他の4指標は対象外にします。空回答は正確性0、回答はあるが模範回答が無い場合は正確性も対象外です。

エージェントの対象外条件

期待する呼び出しも禁止ツールも指定が無い場合、ツール正確性は対象外です。参照回答方式で期待回答が無い場合、条件方式で条件または根拠判定が判定不能場合は、タスク完遂率・最終出力の品質（または回答品質）が対象外として扱われます。判定不能は、証拠として必要なツール応答が欠けている、または長さの上限を超えて切り詰められている場合に起こります。

呼び出し順を採点に含めるかどうかはcallOrderRequiredで切り替えます。真にすると条件方式でも呼び出し順を確認し、順序が違えば減点します。参照回答方式では、一致した期待呼び出しが2件以上あれば常に順序を確認します。

履歴比較の条件

前回比・過去最高値の比較は同じ計画内で行います。比較候補になるのは、実行状態OK・全件処理済み・ケースエラーなしの実行だけです。少数実行、中断、実行エラー、ケースエラーを含む実行は、比較候補から除外します。候補の中から、総件数・指標別の有効件数・保存した評価条件が一致する実行を抽出します。評価条件は、ケースのスナップショットを正規化して照合します。

比較条件に含む内容	比較条件に含めない内容
質問・タスク、入力パラメータ、模範回答、種類、実行ユーザ	ケースID と表示順
期待呼び出し、禁止ツール、判定条件、targetPathなどの期待値	対象エージェントのバージョン
採点モデルID、EmbeddingsモデルID、採点回数	ナレッジ全文や外部データの完全なスナップショット

ケースID や並び順だけの変更は、内容比較の材料にしません。対象バージョンは改善を測るために変える対象であるため、比較条件から除いています。添付ファイルの期待値メタデータは照合材料に含まれますが、外部リソースの同一性そのものを保証するものではありません。条件スナップショットが保持期限などで失われた実行は比較条件を確認できないため、比較対象外です（スコア自体は表示できます）。前回比は「今回値－比較対象値」で求め、過去最高・過去最低は指標ごとに求めるため、各指標の最高値が同じ一つの実行で達成されたとは限りません。

改善提案・権限区分

改善提案は、低スコアの材料に件数の上限があるため、提案が全ケースの問題を網羅するとは限りません。生成に失敗しても、確定済みの評価結果を提案側の失敗によってERRORへ変更することはありません。既にOKまたはRUNNINGの提案がある場合、通常の開始要求では重複生成しません。

HTMLレポートの明細が削除済みの場合、保持している集計値だけからは、失われた判定理由を復元できません。

権限	できること
参照権限	計画・履歴・結果・推移・結果レポートの参照
管理権限	データセット画面の表示、計画やデータセットの変更、評価の開始・中断・再開・再実行・再採点、評価メモの変更、生成、改善提案の生成と参照、添付操作と参照

評価の操作権限と、ケースを実行するユーザの業務権限は別です。実行ユーザを指定すると、その業務権限で測定できますが、評価結果の参照権限が同じユーザへ自動的に付与されるわけではありません。

設定リファレンス

本章では、AIモデル管理の画面項目と、トレースの外部送出設定ファイルの項目を表で示します。項目ごとに用途、未設定時の動作、設定値や制限を確認できます。エージェント定義・ナレッジベース・ツール・スキル・ガードレールの設定項目は、それぞれの章（「[ナレッジベース](#)」～「[ガードレール](#)」）を参照してください。インポート・エクスポートの操作画面と資材の形式は「[インポート・エクスポートできる対象](#)」、手順と注意事項は「[インポート／エクスポート](#)」を参照してください。精度評価のデータセットは「[設定値](#)」、監査ログの CSV エクスポートは「[監査ログ](#)」を参照してください。

AIモデル管理

サイトマップから「Copilot」→「AIモデル」→「AIモデル管理」を開きます。エージェントとナレッジベースが利用するモデルを、テナントごとに登録します。モデルへの接続先や API キーはここでは設定せず、生成AI連携ドライバで設定します（「[生成AIサービスとモデル](#)」）。

基本情報・プロバイダモデル			
設定項目	目的・必要になる条件	未設定時の動作	設定のしかた・判断のヒント・制限
ID	AIモデルを識別する ID。登録時に必要	自動採番された値を使用	必要に応じて登録前に変更する。登録後は変更できない
モデル名	一覧やモデル選択画面に表示する名前。登録時に必要	登録できない	利用者が識別できる名前を入力する。既定では「プロバイダモデルIDと同じにする」が選ばれており、プロバイダモデルID と同じ値が入る
ドライバ種別	接続に使う生成AI連携ドライバを指定。登録時に必要	プロバイダモデルを指定できず、登録できない	「OpenAI」「OpenAI互換サービス」「Azure OpenAI」「Amazon Bedrock」から選ぶ。選択肢は導入済みモジュールに依存し、登録後は変更できない
プロバイダモデルID	プロバイダが定めるモデルの識別子。登録時に必要	登録できない	ドライバ種別を選んでから指定する。利用するモデルの提供元の情報で値を確認する。登録後は変更できない
モデル種別	モデルの用途をテキスト生成・埋め込みから指定。登録時に必要	登録できない	プロバイダモデルID に応じて自動設定される場合がある。エージェントで選ぶモデルにはテキスト生成を指定する。登録後は変更できない
コンテキストウィンドウ	1 回に処理できる入力と出力の合計トークン数。登録時に必要	登録できない	提供元の仕様に沿って入力する。自動コンテキスト圧縮の判定にも使われる（「 エージェント定義 」）

パラメータ仕様・詳細設定			
設定項目	目的・必要になる条件	未設定時の動作	設定のしかた・判断のヒント・制限
<code>max_tokens</code>	生成するトークン数の上限を定めるパラメータ。常に有効	上限を指定しないとパラメータ仕様を確定できない	利用するモデルが許容する最大値を「上限」に入力する。「下限」は 1 に固定。利用側で指定できる範囲はここで定める
<code>temperature</code>	応答のランダム性を調整できるようにする場合に有効化	無効のままなら実行時に送信されない	有効にする場合は、提供元が許容する最小値と最大値を「下限」「上限」に指定する
<code>top_p</code>	サンプリング範囲を調整できるようにする場合に有効化	無効のままなら実行時に送信されない	有効にする場合は、提供元が許容する最小値と最大値を「下限」「上限」に指定する
推論を無効にする	ツール呼び出しと推論を併用できないテキスト生成モデルで使用	無効のままなら推論を禁止しない	GPT-5.6 などツールを使う場合に有効にする。利用側の実行は推論なしになり、推論の段階は選べない

利用側で調整できるパラメータは、有効にしたものだけです。エージェント定義では `temperature` と `max_tokens` を調整できます（「[エージェント定義](#)」）。設定できる範囲はモデルによって異なるため、値は提供元の仕様で確認します。

トークン単価			
設定項目	目的・必要になる条件	未設定時の動作	設定のしかた・判断のヒント・制限
適用開始日	単価の適用期間を指定。単価を登録する場合に必要	その期間の単価を登録できない	「期間を追加」から設定する。価格変更時は新しい適用開始日の期間を追加する
「通常」の入力・出力・キャッシュ単価	実行コストの計算に使用	単価未設定として扱われ、コストを正しく集計できない	USD／100 万トークンで指定する。値は提供元の価格表で確認する
「Long Context」のしきい値・入力・出力・キャッシュ単価	長文コンテキストに別単価があるモデルで使用	Long Context の単価を適用しない	提供元に長文コンテキスト料金がある場合に、しきい値と単価を指定する

単価はダッシュボードやトレースのコスト計算に使われます。

動作保証モデル以外も登録できますが、動作・性能は保証されません（「[生成AIサービスとモデル](#)」）。

登録済みのモデルは、エージェントやナレッジベースから参照されていても削除できます。

トレースの外部送出設定ファイル

トレースを外部の監視基盤へ送出する場合は、`im-copilot-otlp-export-config.xml`を編集します。`enabled`を有効にしない場合、トレースは外部に送出されません。「送出先URL」「認証方式」「追加HTTPヘッダ」はテナント環境セットアップで設定します（「[トレースの外部送出設定 \(OTLP\)](#)」）。以下の各項目を省略した場合は、表の既定値が使われます。

区分	設定項目	目的・設定する条件	既定値	判断のヒント・制限
送出	<code>enabled</code>	外部送出を有効にする	<code>false</code>	外部送出を使う場合は <code>true</code> にする
バッチ	<code>max-queue-size</code> （キューの最大長）	送出待ちのスパンを保持する件数を調整する	2048	1 以上
バッチ	<code>max-export-batch-size</code> （1 回の送出件数）	監視基盤へ一度に送る件数を調整する	512	1 以上。送出先が過負荷になる場合は下げる
バッチ	<code>schedule-delay-millis</code> （送出間隔、ミリ秒）	バッチを送る間隔を調整する	5000	0 以上。送出頻度を下げる場合は伸ばす
バッチ	<code>export-timeout-millis</code> （送出タイムアウト、ミリ秒）	1 回の送出の待機時間を制限する	30000	0 以上
接続	<code>connect-timeout-millis</code> （接続タイムアウト、ミリ秒）	送出先への接続を待つ時間を制限する	10000	1 以上
接続	<code>request-timeout-millis</code> （リクエストタイムアウト、ミリ秒）	送出リクエストの応答を待つ時間を制限する	10000	1 以上
接続	<code>retry-max-attempts</code> （試行回数の上限）	最初の送出を含めた試行回数を制限する。送出に失敗した場合は、この回数に達するまで再試行する	5	2～5。1 は指定できないため、送出に失敗すると必ず 1 回以上再試行する
接続	<code>retry-initial-backoff-millis</code> （再試行の初期待機、ミリ秒）	最初の再試行までの待機時間を調整する	1000	1 以上
接続	<code>retry-max-backoff-millis</code> （再試行の最大待機、ミリ秒）	再試行までの待機時間の上限を定める	5000	1 以上
接続	<code>retry-backoff-multiplier</code> （再試行間隔の倍率）	再試行間隔を増やす倍率を調整する	1.5	0 より大きい
プロキシ	<code>host</code> （プロキシのホスト）	プロキシ経由で送出する場合に接続先を指定する	未設定	プロキシを使わない場合は設定不要
プロキシ	<code>port</code> （プロキシのポート）	プロキシ経由で送出場合にポートを指定する	未設定	プロキシを使わない場合は設定不要

バッチ・接続・プロキシの各項目は、設定ファイルでのみ変更できます。

業務に組み込む

項目

- エージェントの呼び出し方
- IM-LogicDesigner の「エージェント実行」タスク
- IM-BPM から呼ぶ
- Java から呼ぶ
- バージョンの更新と進行中案件

エージェントの呼び出し方

エージェントをIM-LogicDesignerやIM-BPMから呼び出すには、エージェント定義の用途で「ロジックフロータスク」または「BPMタスク」を ON にします。呼び出し元の画面でエージェントを選ぶときは、その用途を ON にしたエージェントだけが候補に表示されます。

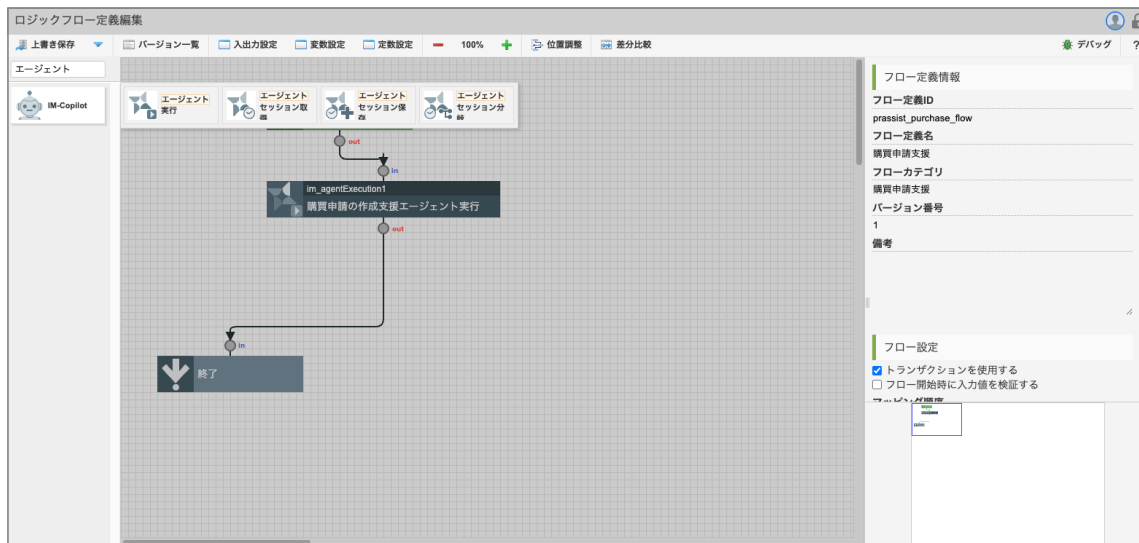
ただし、用途は候補を絞り込むための設定であり、実行時のアクセス制御ではありません。例えばIM-LogicDesignerの「エージェント実行」タスクでは、用途を OFF にしても、すでにタスクに設定したエージェントの実行は止まりません。誰がエージェントを実行できるかは、呼び出し元のロジックフローやワークフローの側で制御します。

同期実行には、呼び出し元のタイムアウトが影響します。エージェントの実行に時間がかかる場合、呼び出し元のタイムアウト設定によっては、応答が返る前に処理が打ち切られることがあります。

エージェントを直接呼び出す公開 REST API は提供されません。外部システムから呼び出す必要がある場合は、IM-LogicDesignerを経由します。「エージェント実行」タスク（「IM-LogicDesigner の「エージェント実行」タスク」）を含むロジックフローを作成し、そのロジックフローを外部へ公開して呼び出します。

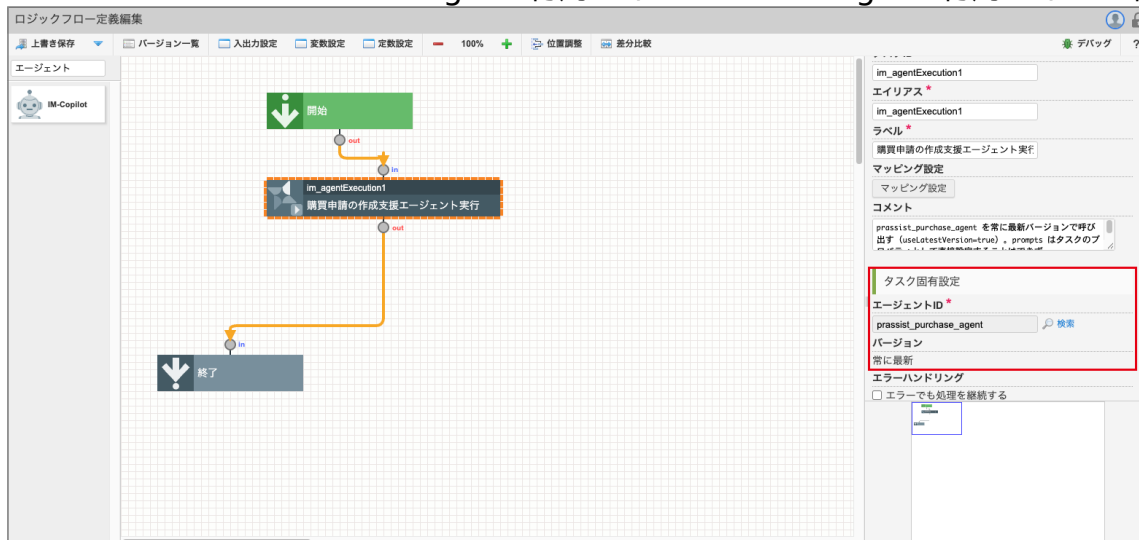
IM-LogicDesigner の「エージェント実行」タスク

「エージェント実行」タスクは、IM-LogicDesigner のパレットのカテゴリ「IM-Copilot」から配置します。このカテゴリには、既存の IM-Copilot タスク（チャット、画像生成、ベクトルDB 操作など）も同居します。



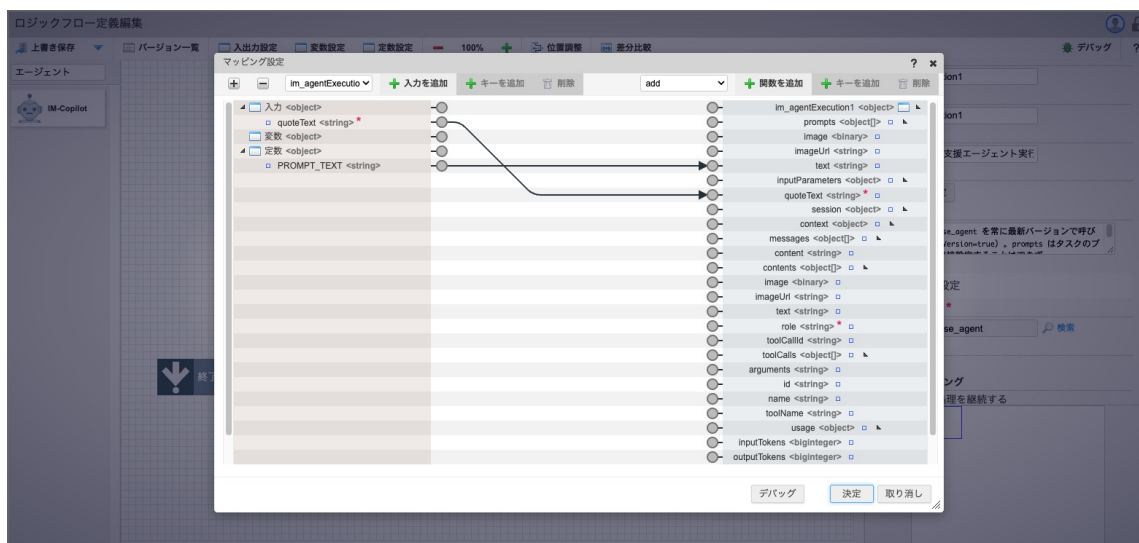
図：パレットの「エージェント実行」タスクと、配置後のキャンバス上のフロー。

エージェントID と、呼び出すバージョンは、タスクのプロパティの「タスク固有設定」で指定します（図の赤枠）。マッピング設定で指定する項目ではありません。エージェントID の検索アイコンから「エージェントを選択」画面を開き、エージェントと、使用するバージョン（「常に最新のバージョンを利用する」または特定のバージョン番号）を選びます。この画面には、用途「ロジックフロータスク」が ON で、公開中のエージェント・バージョンだけが表示されます。選んだバージョンが後から非公開になった場合は実行時にエラーになり、「常に最新のバージョンを利用する」を選んだ場合は、公開中のバージョンのうち最大の番号が使われます。



図：タスクのプロパティ。「タスク固有設定」（赤枠）で、エージェントID とバージョン（「常に最新」）を指定する。

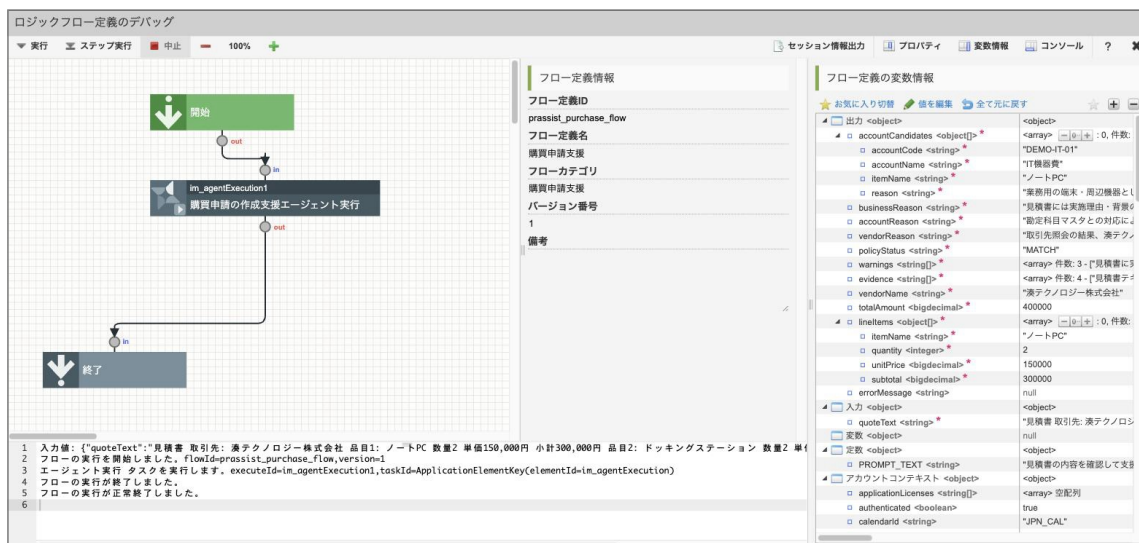
入力には、`prompts`、`inputParameters`、`session` があります。`prompts` の各要素には、`text`・`imageUrl`・`image` のうち、いずれか 1 つを指定します。`prompts` に要素を 1 つも指定しない場合はエラーが発生します。`inputParameters` は、エージェント定義に入力スキーマが設定されている場合にだけ生成されます。



図：マッピング設定画面。`prompts` や `inputParameters` に、開始タスクの入力や定数をマッピングする。

出力には、`result` と `session` があります。エージェントの応答は `result` に入ります。エージェント定義で構造化出力を設定している場合は、出力パラメータの値が `result` の `outputParameters` に入ります。

入力・出力の各項目の型や詳細は、「IM-LogicDesigner仕様書」の「エージェント実行」を参照してください。



図：フロー定義のデバッグ実行結果。「フロー定義の変数情報」の「出力」に、構造化出力のスキーマに沿った `outputParameters` の値が展開される。

注意

IM-LogicDesignerのデバッグ実行からエージェントを呼び出した場合も、実行種別は「テスト」ではなく「本番」として記録されます。デバッグ実行であっても、AI モデルの呼び出しによる課金や、ツール（外部システムへの登録・更新など）の実行は実際に行われます。また、デバッグ実行の結果は、運用ダッシュボードや経営ダッシュボードの本番実行の集計にも含まれます。

コラム

「エージェント実行」タスク自体は、セッションを保存しません。会話の続きを扱いたい場合は、エージェントセッション取得・保存・分岐の各タスクを組み合わせます。保存タスクは、まだ保存されていない部分だけを追記する動作であり、保存済みの履歴と渡した履歴の先頭が一致しないと失敗します。分岐タスクは、分岐元のセッションを残したまま新しいセッションを作ります。

また、自動コンテキスト圧縮を働かせるには、`session.context.usage` を渡す必要があります。渡さない場合、圧縮は行われません。

IM-BPM から呼ぶ

用途で「BPMタスク」を選んだエージェントは、IM-BPM のプロセスから呼び出せます。IM-BPM が未導入の環境では、この用途そのものが選択肢に表示されません。

IM-BPM のプロセスデザイナーには、専用の「AccelAgentタスク」があります。タスクには、連携先のエージェントを指定する連携ID、利用するバージョン（常に最新を使用するか、バージョン番号を指定するか。EL 式での動的指定も可能）、エージェントへ渡す入力データを設定します。実行結果を後続のタスクで使う場合は「結果変数を格納する」を有効にし、結果変数名（例: `result`）を指定すると、以降のタスクから `${result}` のような EL 式（「IM-BPM 仕様書」の「EL式」）で参照できます。

入力データのうち、名前を `message` にした項目は、エージェントへのメッセージとして渡されます。それ以外の項目は、エージェント定義のインプットパラメータとして渡されます。`message` を設定しない場合は、空のメッセージで実行されます。

バージョン番号を指定した場合も、そのバージョンが公開されていないとエラーが発生します。結果変数には、エージェント定義で構造化出力を設定している場合はその値が、設定していない場合は応答の本文が格納されます。

エージェントの実行がエラーやガードレールによる遮断で終わった場合は、プロセスの実行がエラーで終了します。また、IM-LogicDesignerから呼び出す場合と同じく、実行種別は「本番」として記録され、AI モデルの呼び出しによる課金やツールの実行も実際に行われます。

Java から呼ぶ

Java のプログラムからも、エージェント定義を指定してエージェントを呼び出せます。`AgentManagementServiceFactory` から取得した `AgentManagementService` でエージェントを読み込み、実行します。

`getAgent` メソッドは、バージョンの公開状態を問わずにエージェントを読み込みます。IM-LogicDesignerやIM-BPMのタスクと同じく公開中のバージョンだけを実行したい場合は、`resolvePublishedVersion` メソッドで解決したバージョン番号を指定して読み込みます。

クラスやメソッドの詳細は、「[jp.co.intra_mart.foundation.copilot.agent パッケージの API ドキュメント](#)」を参照してください。

バージョンの更新と進行中案件

特定のバージョン番号を指定して呼び出している箇所がある状態で、そのバージョンを非公開にすると、呼び出しは失敗します。IM-LogicDesignerの「エージェント実行」タスクとIM-BPMの「AccelAgentタスク」は、どちらもタスクを実行する時点でバージョンの公開状態を確認するためです。IM-LogicDesignerではフローの実行が、IM-BPMではプロセスの実行がエラーで終了します。

「常に最新のバージョンを利用する」を選んだ呼び出しは、実行する時点で公開中のバージョンのうち最も大きい番号を使います。そのため、公開中のバージョンが1つでも残っていれば失敗しません。ただし、新しいバージョンを公開すると、すべての呼び出し元が次の実行から新しいバージョンを使います。インプットパラメータや構造化出力の項目を変えた場合は、後続の処理が想定どおりに動くか、公開前に確認します。

バージョン番号を指定して呼び出しているエージェントを新しいバージョンに切り替える場合は、次の順に作業します。

1. 新しいバージョンを公開します。古いバージョンは、この時点では非公開にしません。
2. 古いバージョンの番号を指定しているIM-LogicDesignerのフローとIM-BPMのプロセスを洗い出し、指定を新しいバージョンに変更します。
3. 変更した呼び出し元を実行し、新しいバージョンで想定どおりに動くことを確認します。
4. 古いバージョンを呼び出す処理が残っていないことを確かめてから、古いバージョンを非公開にします。

IM-BPMでは、進行中の案件にも注意します。AccelAgentタスクのバージョン指定はプロセス定義に含まれるため、案件がどのバージョンを呼び出すかは、その案件が使っているプロセス定義で決まります。古いバージョンを指定したプロセス定義で進んでいる案件が残っている間は、古いバージョンを公開したままにしておき、それらの案件がAccelAgentタスクを通過してから非公開にします。

エージェント定義をインポートすると、インポート先で公開中だったバージョンが非公開に変わることがあります（警告は表示されます）。稼働中の環境へ取り込む際は、インポート後に公開状態を確認する作業とセットで行います（「[インポートで何が起きるか](#)」）。

項目
■ 何が見える画面か
■ トレース一覧・詳細の読み方
■ 評価結果からトレースへ
■ コストの見方
■ 外部の監視基盤で追う
■ 保持期間と閲覧権限
■ 読み換えやすい点

何が見える画面か

この画面は、エージェントの実行を一覧・絞り込みし、気になる 1 件をスパン単位まで掘り下げて原因を確認するためのものです。典型的には、次のような場面で使います。

- 運用ダッシュボードで異常の兆候（エラー率の上昇、特定エージェントの遅延、ガードレールの発動など）に気づいたとき（「[トレースへの掘り下げ](#)」）
- 精度評価で失敗ケースを見つけ、指標だけでは分からない中身を確認したいとき（「[評価結果からトレースへ](#)」）
- 利用者から個別の問い合わせがあり、特定の実行を直接調べたいとき

サイトマップから「Copilot」→「Accel Agent」→「ワークスペース」を開き、画面左のサイドナビゲーションの「トレース」から移動します。

トレースは、1 回の実行を構成する個々の処理単位（スパン）を、親子関係を持つ木構造として記録したものです。例えば「実行全体」のスパンの中に「イテレーション」のスパンがあり、その中に「モデル呼び出し」「ツール呼び出し」などのスパンが入ります。スパンの種別には、実行全体、イテレーション、モデル呼び出し、ツール呼び出し、ナレッジ検索、ガードレールなどがあります。

エージェントの実行に加えて、エージェント定義を経由しないIM-Copilotの生成AIの呼び出し（チャット アクション・埋め込み アクションなど）も、プロセス種別「直接実行」として記録されます。

画面は一覧と詳細の 2 段構成です。トレース一覧は、原因そのものは示さず、「いつ・どのエージェントが・どれだけ時間とコストを使い・どう終わったか」を並べて、掘り下げる実行を選ぶための画面です。トレース詳細は、選んだ 1 件について、スパンの木・時系列・中身まで掘り下げて原因を読むための画面です。

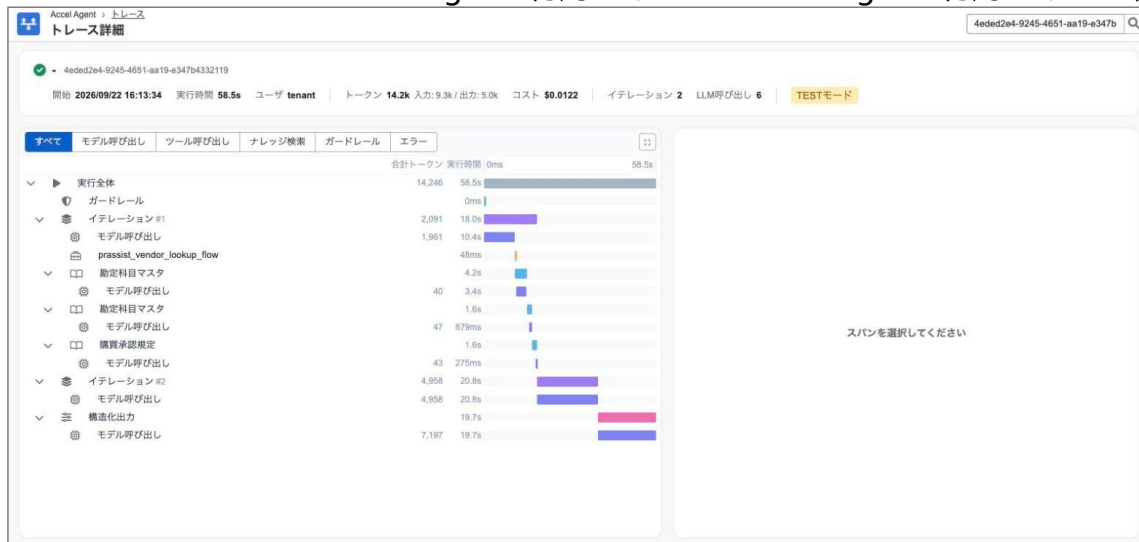
トレース一覧・詳細の読み方

一覧の列	内容
ステータス・開始時刻・プロセス種別・プロセス名・バージョン	実行の基本情報
実行時間	1 秒未満はミリ秒、それ以上は秒で表示
LLM 呼び出し回数	この実行で LLM を呼んだ回数
コスト・実行種別・エラー種別	コストは単価未設定の場合「-（ハイフン）」。エラー種別は値があるときだけバッジ表示

イテレーション回数は列としても絞り込みとしても持ちません。イテレーション回数は実行ストラテジによって「1 回のイテレーションで LLM を何回呼ぶか」が変わり、繰り返しの多さを比べる指標として意味が揺れるためです。この代わりに、ストラテジによらず同じ意味を持つ LLM 呼び出し回数に一本化しています。

一覧の絞り込みは、期間・プロセス種別・エージェント・バージョン（エージェントを選んだときだけ表示）・ステータス（「すべて」「エラー」「正常」の 3 つ。初期表示は「すべて」）・エラー種別（ステータスを「エラー」にしたときだけ表示）・実行種別・所要時間（下限・上限）・使用モデル・LLM 呼び出し回数（下限・上限）・要注意実行（「ツールエラーあり」「出力リトライあり」を独立した ON/OFF で指定し、両方 ON なら両方に該当する実行だけを絞り込む）です。並べ替えができるのは、開始時刻（既定・降順）・実行時間・LLM 呼び出し回数・コストの 4 列だけです。1 ページは 20 件です。

詳細画面は、上から、実行全体の概要（サマリ帯）、スパンツリー（ウォーターフォールを兼ねる）、スパン詳細の 3 段で構成されます。サマリ帯には実行全体の状態・トークン数・コスト・イテレーション数・LLM 呼び出し回数が並びます。スパンツリーは親子構造とウォーターフォール形式の時系列を兼ね、初回トークンが返るまでの時間（TTFT）はモデル呼び出しスパンを選んだときにスパン詳細で確認できます。プロンプト・応答・ツールの入出力は折りたたみで表示され、プロンプトだけは既定で閉じています（本文が長く、開いたままだと後続の項目が画面から隠れるためです）。



図：トレース詳細画面。モデル呼び出し・ツール呼び出し・ナレッジ検索のスパンが、親子構造とウォーターフォールで並ぶ。

トレースからは、そのエージェント定義の編集画面へ逆リンクできますが、これはエージェント経由の実行の場合だけです。エージェント定義を持たない実行では、リンクの代わりに理由が表示されます。逆リンクの文言には選択中のスパンに応じたセクション名が表示されますが、移動先はエージェントID とバージョンに対応する編集画面です。セクションは指定されません。

評価結果からトレースへ

精度評価の結果画面からは、対応するトレースへ辿ることができます。指標だけでは分からない失敗の内容を、トレースの詳細で確認します。

コストの見方

トークンの単価は AI モデル管理で設定します。精度評価では、対象エージェントの実行コストと採点コストが別々に記録されます。対象コストは対象エージェント自身が使うモデルの単価から、採点コストは採点モデルと Embeddings モデルの単価から、それぞれ算出されます。データセットの自動生成や改善提案の呼び出しにも別途コストがかかるため、評価実行の一つの金額を総費用として扱うことはできません。

コラム

単価が未設定のモデルによる実行は、コストが 0 ではなく「- (ハイフン)」と表示され、金額の集計から除外されます。画面には除外された件数の注記が表示されますが、トレース一覧はトレース単位、経営ダッシュボードはスパン単位で件数を数えているため、両者の件数は一致しません。単価ありと単価なしの実行が混ざっている場合、一覧の件数には現れず、金額だけが実態より小さく見えます。

金額は USD で集計・表示されます。画面右上の「1 USD =」の欄に通貨単位（既定は「¥」）とレートを入力すると、画面上の金額をその通貨に換算して表示できます。運用ダッシュボードと経営ダッシュボードでも、同じ欄で換算できます。換算は表示のための概算で、集計は USD のまま行われます。入力したレートと通貨単位は保存されず、画面を開き直すと USD の表示に戻ります。為替レートは日々変動するため、期間の集計を 1 つのレートで換算した値は、あくまで概算として扱ってください。

外部の監視基盤で追う

「[トレースの外部送出設定 \(OTLP\)](#)」でトレースの外部送出を有効にした場合、外部の監視基盤にはトレース（スパン）だけが送られます。メトリクス、ログ、監査ログは送出の対象に含まれません。プロンプトや応答の本文も送られず、送られるのは属性のみです。どこまで細かい属性が送られるかは、内部のダッシュボードで見るとの使い分けの判断材料です。

保持期間と閲覧権限

スパン詳細（プロンプト・応答・ツールの入出力・検索クエリなど、実行内容の本文）は、保持期間を過ぎると参照できなくなります。実行の記録自体は一覧に残りますが、詳細画面を開くと本文の代わりに「保持期間超過のため表示不可」という案内に置き換わります。

トレースを参照する権限は、一覧と詳細で分かれています。一覧を見られる利用者は、実行内容の本文（プロンプト・応答・ツールの入出力など）も含めて、テナント内の全実行を参照できます。「一覧だけ見せて中身は見せない」という権限の分け方は現在できないため、権限を割り当てる際はこの点を踏まえます。

読み違いやすい点

- トレース一覧のコスト列の「- (ハイフン)」は 0 ではなく、単価未設定で金額を算出できないことを示します。実際にコストが 0 で終わった実行（LLM に到達する前に終了した実行など）は「\$0.0000」と表示され、区別されます。
- サマリ帯に出る「イテレーション」はトレース全体の集計値、スパンツリーの各イテレーションスパンに付く番号は個々のイテレーションの通し番号です。ツリーを絞り込んで表示行数が減っても、サマリ帯の値は変わりません。

本章は、開発環境で作った定義を別のテナントや別の環境へ移す作業を扱います。順序を誤ると動作せず、取り込み方によっては既存のデータが失われるため、本章の手順に沿って作業します。対象は定義体のみであり、精度評価のデータセットは「[精度評価](#)」、監査ログの CSV エクスポートは「[監査ログ](#)」を参照してください。

本章の読者は「Accel Agent 管理者」です。インポート・エクスポートの画面はいずれも管理権限を要するため、参照のみのロールでは画面にすら到達できません。本番環境への取り込みを運用担当が行う場合も、本章の手順に沿えば作業を完了できます。

- 項目
- インポート・エクスポートできる対象
 - 共通の進め方と事故への備え
 - インポート先の前提とインポートの順番
 - エクスポートの注意点
 - インポートで何が起きるか
 - インポート後に確かめること

インポート・エクスポートできる対象

インポート・エクスポートの対象は、エージェント定義、ナレッジベース、ロジックツール、MCP、スキル、AIモデルです。いずれも、それぞれの管理画面にある「インポート」「エクスポート」から操作します。対象ごとに、資材の形式、含まれないもの、取り込み方式、秘匿情報の有無が異なります。

対象	操作する画面	エクスポートされる資材	取り込み方式
エージェント定義	エージェント管理	<code>agent_data.json</code>	同じ ID のものを上書き
ナレッジベース	ナレッジベース管理	<code>knowledge_data.zip</code>	差分同期（資材に無い既存アイテムは削除）
ロジックツール	ツール管理	<code>logic_tool_data.json</code>	同じ ID のものを上書き
MCP	ツール管理	<code>mcp-server.json</code>	同じ ID のものを上書き
スキル	スキル管理	<code>skill_data.zip</code>	洗い替え（1 件でも不正な内容があると全体が失敗）
AIモデル	AIモデル管理	<code>ai_model_data.json</code>	同じ ID のものを上書き

資材に含まれないものや秘匿情報の扱いは「[エクスポートの注意点](#)」、取り込み時の挙動は「[インポートで何が起きるか](#)」で確認します。

共通の進め方と事故への備え

どの対象も、資材をアップロードしてから内容を確認し、実行するという 2 段階の方式で進みます。確認画面で状態を示す単位と、示される状態は対象によって異なります。

- ロジックツール・MCP：1 件ごとに「新規」「更新」「変更なし」を示します。
- ナレッジベース：カテゴリごとに「新規」「更新」「変更なし」を、ナレッジベースごとに「新規」「更新」を示します。ナレッジベースは内容が同じでも、インポート先に同じ ID のものがあれば「更新」と表示されます。

取り込みを実行する前には、必ずインポート先を先にエクスポートして退避します。取り込み後に元の状態へ戻す手段は用意されていないため、これが唯一の備えです。ナレッジベースは、退避のためのエクスポートそのものが失敗することがあるため、先にアイテムの状態を確認しておきます。MCP を含む資材には認証情報が含まれるため、秘密として保管し、受け渡しします。

インポート先の前提とインポートの順番

画面の外で済ませておく事項は、何をインポートするかによって変わります。すべてが常に必要なわけではないため、対象に応じて確認します。

事前に済ませておく事項	必要になる場合	手順
テナント環境セットアップのベクトルデータベース接続	ナレッジベースをインポートする場合	「 ベクトルデータベース接続設定 」
生成AI 連携ドライバ設定	AIモデルをインポートする場合	「 生成AI連携ドライバ設定 」
ジョブの設定	ナレッジベースをインポートする場合	「 ジョブネットの作成と有効化 」
IM-BPM の導入	用途に「BPMタスク」を含むエージェント定義をインポートする場合	「 IM-Juggling プロジェクトの編集 」
OAuth プロバイダ設定	OAuth 認証を使う MCP サーバをインポートする場合	「 OAuth の認証プロバイダを登録する 」
ロジックフローのインポート	ロジックツール、またはロジックフロータスクを使うエージェント定義をインポートする場合	IM-LogicDesigner仕様書 の「 インポート・エクスポート 」

そのうえで、AIモデル、次にナレッジベース・スキル・MCP・ロジックツール、最後にエージェント定義、という順でインポートします。ロジックツールは、参照するロジックフローをIM-LogicDesignerでインポートした後に取り込みます。参照先のロジックフローがインポート先に無くても取り込みは止まらず、インポート結果の詳細に警告が表示されます。エージェント定義がAIモデルを解決できないまま取り込まれた場合、取り込み自体は止まらず警告として記録されますが、そのエージェントはまったく動作しません。これに対して、ナレッジベース・ツール・スキル・MCP が未解決の場合は、その部品を使う場面だけが失敗します。

ナレッジベースをインポートする場合は、扱いが異なります。埋め込みモデル（AIモデル）を解決できないと、そのナレッジベースのインポート自体がエラーで止まり、取り込めません。

！ 注意

インポート先で定義を手作業で作直すことは避けてください。エージェントID・AIモデルID・ナレッジベースID には新規作成時に自動で採番された値が入るため、同じ値を入力し直さずに作り直すと ID が変わり、参照が切れます。ツールID と Skill ID は、環境をまたいで同じ値になるようそろえてください。

エクスポートの注意点

資材ごとに、含まれる内容と注意点が異なります。

- **エージェント定義**: 参照しているナレッジベース・ツール・スキル・AIモデルは含まれません。インポート先では、ID が一致するものを参照します。
- **ナレッジベース**: ベクトルデータ本体が含まれるため、容量が大きくなります。配下のアイテムに、待機中・処理中・エラー・メタデータ同期失敗のいずれかが1件でも含まれる場合は、エクスポートできません。
- **ロジックツール**: 参照するロジックフローの定義は含まれません。ロジックフローはIM-LogicDesignerでエクスポートします。
- **MCP**: 認証方式が API キーの場合は、管理時・実行時の API キーが平文で含まれます。追加の HTTP ヘッダも、設定した値のまま含まれます。これらを含む資材は、秘密情報として保管し、受け渡します。OAuth の場合は、使用する OAuth プロバイダの指定だけが含まれ、認証情報は含まれないため、インポート先でも同じ OAuth プロバイダを設定しておきます（「[MCPサーバに接続する](#)」）。
- **AIモデル**: 接続先や API キーなどの認証情報は含まれません。認証情報は生成AI連携ドライバ設定で管理するため、インポート先でも同じ設定をしておきます（「[生成AI連携ドライバ設定](#)」）。

インポートで何が起きるか

取り込み時の挙動は、対象ごとに異なります。

ナレッジベースのアイテムは差分同期です。資材に含まれていない既存のアイテムは削除されます。インポートの確認画面では、削除されるアイテムは表示されません。

取り込む前に、インポート先のナレッジベースをエクスポートして退避してください。意図しないアイテムが削除された場合は、退避した資材をインポートし直すと、退避した時点の状態に戻せます。

ベクトルデータ（チャンク）だけは差分同期の対象外です。資材にベクトルファイルが含まれるかどうかに関わらず、そのナレッジベースの既存チャンクをいったん全件削除してから、資材の内容で登録し直します。取り込み中は、該当ナレッジベースの検索結果が一時的に欠ける、または空になることがあります。

無効化されているナレッジベースを、インポート先で有効かつ利用中（いずれかのエージェントに割り当て済み）のナレッジベースへ取り込もうとすると、行エラーになり取り込めません。

スキルは洗い替えであり、1件でも不正な内容が含まれていると、取り込み全体が失敗します。それ以外の対象は、同じ ID のものを上書きする方式です。

エージェント定義の取り込みでは、次のような場合に警告が表示されます。インポート先で公開中だったバージョンが非公開に変わるケース（稼働中の環境では、呼び出し元に即座に影響します。「[バージョンの更新と進行中案件](#)」）、参照を解決できないケース（該当箇所の呼び出しだけが失敗する）、AIモデルを解決できないケース（エージェントの復元自体が失敗し、まったく動かなくなる）です。なお、インポート先ですでに削除されていたバージョンが復活することもあります。これは警告としては表示されません。

用途に「BPMタスク」を含むエージェント定義を、IM-BPM が未導入の環境へ取り込んでも、この用途は保持されます。編集画面ではチェックボックスが表示されなくなるだけで、値は消えません。後から IM-BPM を導入すれば、そのまま呼び出せます。エージェントは公開しない限り呼び出し元から使えない点は、「[業務に組み込む](#)」と共通です。

インポート後に確かめること

インポートが終わったら、次の点を確認します。

- **公開状態**: エージェント定義のインポート結果に、公開中だったバージョンが非公開に変わった旨の警告が出ていないか確認します。該当する場合は、必要なバージョンを公開し直します（「[インポートで何が起きるか](#)」）。
- **参照切れ**: インポート結果に、参照先（AIモデル・ナレッジベース・ツール・スキル）を解決できなかった旨の警告が出ていないか確認します。該当する場合は、不足している資材をインポートするか、エージェント定義の割り当てを見直します。
- **ジョブの設定**: ジョブネットはインポート・エクスポートの対象外です。インポート先でも、必要なジョブネットが作成・有効化されているか確認します（「[ジョブネットの作成と有効化](#)」）。
- **AIモデルの単価**: 単価は資材に含まれてインポートされます。インポート先で利用する契約や料金がインポート元と異なる場合や、単価が未設定のまま取り込まれた場合は、コストが正しく集計されません。単価がインポート先の環境に合っているか確認します（「[AIモデル管理](#)」）。

運用ダッシュボード

本章の読者は運用管理者を想定しており、参照のみのロールで到達できます。画面を開き、表示される数字の意味を本章で確かめながら読み進めてください。

項目

- 何が見える画面か
- KPI の 4 枚
- 推移グラフ
- 内訳と一覧
 - エラー種別内訳
 - スパン種別別 平均所要時間
 - ツール別利用状況
 - ナレッジ検索ゼロヒット
 - 直近エラー一覧
 - エージェント別ランキング
- トレースへの掘り下げ
- 読み違えやすい点

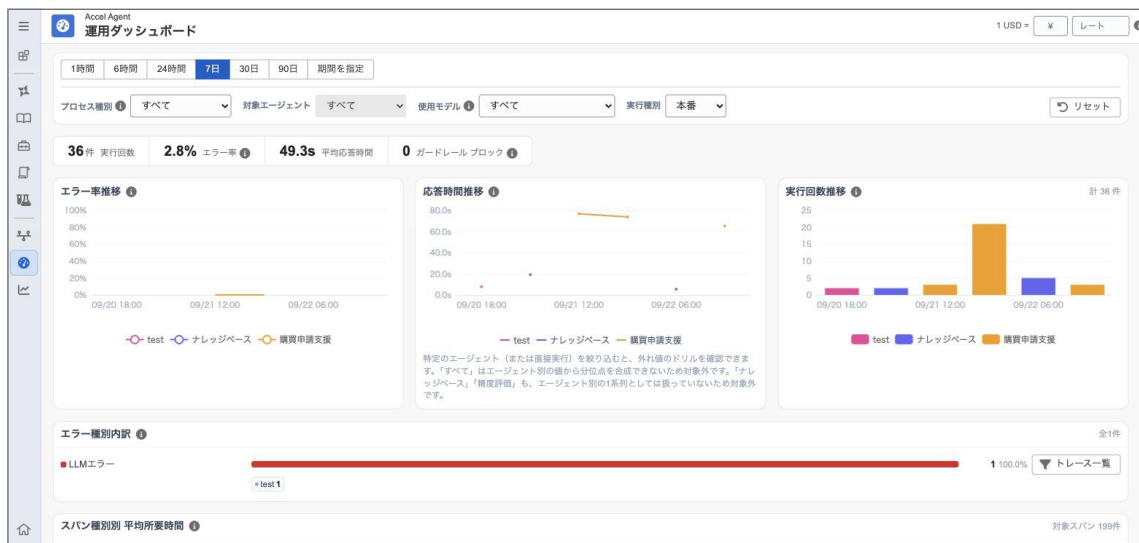
何が見える画面か

この画面は、エージェントの実行を集計して「壊れていないか・遅くなっていないか」（品質・性能）を確認するためのものです。異常の兆候（エラー率の上昇、特定エージェントの遅延、ガードレールの発動など）に気づいたら、トレース一覧・詳細へ条件を引き継いだまま掘り下げる起点です。費用を横断的に分析する画面は「[経営ダッシュボード](#)」であり、本画面の「エージェント別ランキング」に出てくる「1実行あたりコスト」は異常検知の補助情報という位置づけです。

サイトマップから「Copilot」→「Accel Agent」→「ワークスペース」を開き、画面左のサイドナビゲーションの「運用ダッシュボード」から移動します。

期間、実行種別、プロセス種別、対象エージェント、使用モデルで絞り込みます。期間の既定は直近 7 日間で、1 時間・6 時間・24 時間・7 日・30 日・90 日・カスタム（開始・終了を指定）から選べます。

使用モデルの絞り込みは、その実行で実際に使われた全モデル（LLM 呼び出しのモデル名）が対象です。トレースが持つ主モデル（代表の 1 モデル）だけでは判定しないため、1 回の実行の中で一部だけ別モデルヘルパーティングした場合も、この絞り込みに含まれます。



図：運用ダッシュボードの画面上部。絞り込み、4 つの KPI、エラー率・応答時間・実行回数の推移グラフが並ぶ。

KPI の 4 枚

画面上部には、実行回数、エラー率、平均応答時間、ガードレール ブロックという 4 つの KPI が並びます。分母となる実行数が 0 件のとき、エラー率は「0%」ではなく「-（ハイフン）」と表示されます（母数が無い以上、率を定義できないためです）。

推移グラフ

エラー率、応答時間、実行回数の 3 種類について、推移グラフが表示されます。系列の数が多い場合の扱いは指標によって異なります。実行回数は合算できるため、上位以外を「その他」にまとめて棒の高さ（総実行回数）を保ちます。一方、エラー率と応答時間は合算できない値のため、上位以外の系列は「その他」に束ねずに非表示とし、省略した件数を注記で示します。

応答時間推移は、対象エージェントの絞り込み方で見え方が変わります。「すべて」のときはエージェントごとの p90 を色分けした折れ線になり、特定のエージェント（または直接実行）に絞ったときは、その母集団の平均と分位点（p50・p90・p95・p99）の 5 本に切り替わります。

応答時間のグラフに出てくる値の意味は、次のとおりです。

値	意味
平均	すべての実行の応答時間を足して、実行数で割った値。極端に遅い実行が少しあるだけで大きく引き上げられる
p50（中央値）	応答時間を短い順に並べたとき、ちょうど真ん中にある値。半数の実行はこれより速く終わる。利用者が普段感じる速さに近い
p90・p95・p99	それぞれ、実行の 90%・95%・99% がその時間以内に終わる値。残りの 10%・5%・1% は、これより遅い。遅い側の実行がどれだけ遅いかを表す

これらの値は、組み合わせることで、遅さの原因の見当をつけられます。

- **平均が p50 より大きく離れている**: 一部の極端に遅い実行が、平均を引き上げています。トレース一覧を「実行時間」の長い順に並べ替えて、遅い実行を個別に確認します（「[トレース一覧・詳細の読み方](#)」）。
- **p50 は安定していて、p95・p99 だけが上がっている**: 大半の実行は普段どおりで、一部の実行だけが遅くなっています。特定の入力でツールの呼び出し回数が増えている、外部サービスの応答が遅いなど、条件が限られた原因を疑います。
- **p50 を含めて全体が上がっている**: すべての実行が遅くなっています。AIモデルの変更や、AIサービス側の応答の遅れなど、全体に影響する原因を疑います。
- **「すべて」を選んだときのエージェントごとの p90**: エージェント同士の比較に使います。他より p90 が高いエージェントから確認します。

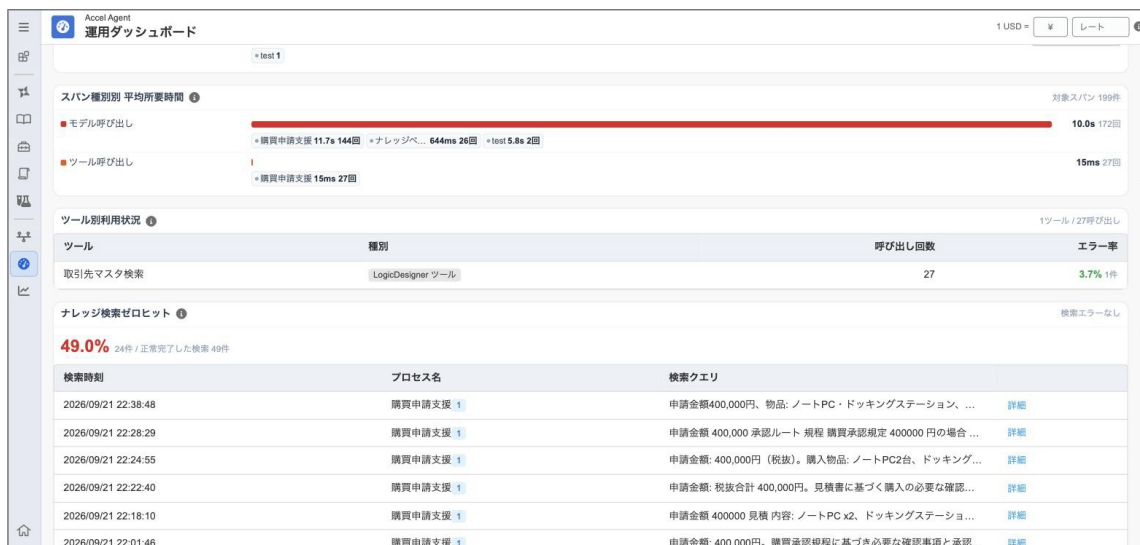
実行数が少ない期間では、p95・p99 は数件の実行だけで決まるため、値が大きくぶれます。例えば実行が 100 件未満なら、p99 はほぼ最も遅い 1 件の値です。上位の分位点は、ある程度の実行数がある期間で見るとようになります。

いずれの推移グラフも、対象期間内に実行が無かった区間は「0%」「0 秒」ではなく、線を途切れさせて空欄で表します。実行が無かったことと、実行はあったが問題が無かったこと（0 の実績）を区別するためです。

内訳と一覧

内訳と一覧は、知りたいことに応じて次のように使い分けます。

知りたいこと	見るところ
失敗がどの種類に偏っているか どの処理で起きているか	「エラー種別内訳」
1 回あたりの時間が長い処理はどれか 呼び出し回数が多く、合計では時間を使っている処理が無い	「スパン種別別 平均所要時間」
特定のツールだけ失敗していないか どのツールがよく使われているか	「ツール別利用状況」
回答の根拠になる文書が見つからないか 空振りした検索は何を問い合わせていたか	「ナレッジ検索ゼロヒット」
いま起きている失敗の中身を 1 件ずつ見たい 新しいバージョンを公開してから失敗が増えていないか	「直近エラー一覧」
どのエージェントを優先して調べるか エージェントごとの状態を並べて比べたい	「エージェント別ランキング」



図：「スパン種別別 平均所要時間」「ツール別利用状況」「ナレッジ検索ゼロヒット」（ゼロヒットの割合と、検索クエリ・発生時刻の一覧）が並ぶ。

エラー種別内訳

エラーで終わった実行を、エラーが起きた処理（エラー種別）ごとに集計します。

- **表示内容**: LLMエラー・ツールエラー・ナレッジエラー・構造化出力エラー・コンテキスト圧縮エラー・その他の件数と、エラー総件数に対する構成比。ガードレールによるブロックは障害ではなく防御動作のため、区切りの下に、検査位置ごとに分けて表示されます。
- **読み方**: 件数の多い種別が、失敗の多い処理です。例えば、LLMエラーが多ければ AI サービスや AI モデルの設定を、ツールエラーが多ければツールの接続先を確認

認めます。

- **操作:** 行を展開すると、エージェント別の内訳が表示されます（件数が多いときは「内訳を表示（N件）」から残りを表示できます）。各種別の「トレース一覧」から、その種別で終わった実行を一覧で確認できます。

スパン種別別 平均所要時間

応答時間の内訳として、どの処理に時間がかかっているかを表示します。

- **表示内容:** 処理の種類（モデル呼び出し、ツール呼び出し、ナレッジ検索、ガードレール、構造化出力、コンテキスト圧縮、検索クエリ書き換え）ごとの、1 回あたりの平均所要時間と呼び出し回数。
- **読み方:** 1 回あたりの時間が長い処理と、呼び出し回数が多い処理の両方を見ます。1 回は短くても何度も呼ばれる処理は、全体への影響が大きくなります。例えば、モデル呼び出しが長ければ AI モデルの応答やプロンプトの長さを、ツール呼び出しが長ければ接続先のシステムの応答を疑います。
- **注意:** 値は 1 回の呼び出しにかかった時間です。1 回の実行あたりの内訳ではないため、合計しても平均応答時間にはなりません。また、実行全体やイテレーションのように、ほかの処理を内側に含むものは、時間を重ねて数えないよう集計から除いています。
- **操作:** 行を展開すると、エージェント別の内訳が表示されます（件数が多いときは「内訳を表示（N件）」から残りを表示できます）。

ツール別利用状況

ツールごとの呼び出し回数とエラー率を表示します。

- **表示内容:** 「ツール」「種別」「呼び出し回数」「エラー率」の表。集計はツール呼び出し 1 回ごとで、実行（トレース）単位ではありません。
- **読み方:** エラー率の高いツールが無いかを確認します。ツールが失敗しても、エラーの内容が LLM に返されて処理が続くことがあるため、実行全体は正常に終わっていることがよくあります。こうした失敗は「直近エラー一覧」には現れないため、ここで確認します。

ナレッジ検索ゼロヒット

ナレッジ検索のうち、参照できる文書が 1 件も見つからなかった割合を表示します。

- **表示内容:** 「N件 / 正常完了した検索 M件」の形の件数と、ゼロヒットになった検索クエリ・発生時刻の一覧。検索自体が失敗したものは分母から除き、除いた件数を別に表示します。
- **読み方:** 一覧の検索クエリを見て、何を探して見つからなかったのかを確認します。ナレッジベースに必要な文書が無い、または検索の条件が合っていない可能性があります。行をクリックすると、そのトレースの詳細に移動します。
- **注意:** ゼロヒットは正常な応答として扱われるため、エラー率にもエラー種別内訳にも現れません。この表示でしか気づけない劣化です。

直近エラー一覧

エラーで終わった実行を、新しい順に 1 件ずつ表示します。

- **表示内容:** 現在の絞り込みに該当する実行を、開始時刻の新しい順に最大 20 件。列は「開始時刻」「プロセス名」「バージョン」「エラー種別」「実行時間」です。
- **読み方:** 個々の失敗の中身を確認する入口として使います。「バージョン」の列から、新しいバージョンを公開してから失敗が増えていないかも確認できます。
- **注意:** ガードレールによるブロックも、実行が中断された事象として含みます。KPI の「エラー率」はブロックを含まないため、件数の考え方が異なります。
- **操作:** 行をクリックすると、そのトレースの詳細に移動します。20 件を超える場合は「全N件中 直近20件」と表示され、「すべて表示」から、現在の期間・実行種別・プロセス種別・対象エージェントの条件を引き継いだトレース一覧で全件を確認できます。

エージェント別ランキング

エージェントごとの状態を並べて比べます。

- **表示内容:** 「エラー率」「平均応答時間」「p95応答時間」「ガードレール ブロック」「1実行あたりコスト」「1実行あたりトークン」「実行数」の列。
- **読み方:** 既定ではエラー率の高い順に並ぶため、上から順に調べます。列見出しをクリックすると、その列で並べ替えられます。

三

運用ダッシュボード

二

一

三

二

一

二

三

二

一

二

三

二

一

二

三

二

一

二

三

二

一

二

三

二

一

二

三

二

一

二

三

二

一

二

三

二

一

二

三

二

一

二

三

二

一

二

三

二

一

二

三

二

一

二

三

二

一

二

三

二

一

二

三

二

一

二

三

二

一

二

三

二

一

二

三

二

一

二

三

二

一

二

三

二

一

二

三

二

一

二

三

二

一

二

三

二

一

二

三

二

一

二

三

二

一

二

三

二

一

二

三

二

一

二

三

二

一

二

三

二

一

二

三

二

一

二

三

二

一

二

三

二

一

二

三

二

一

二

三

二

一

二

三

二

一

二

三

二

一

二

三

二

一

二

三

二

一

二

三

二

一

二

三

二

一

二

三

二

一

二

三

二

一

二

三

二

一

二

三

二

一

二

三

二

一

二

三

二

一

二

三

二

一

二

三

二

一

二

三

二

一

二

三

二

一

二

三

二

一

二

三

二

一

二

三

二

一

二

三

二

一

二

三

二

一

二

三

二

一

二

三

二

一

二

三

二

一

二

三

二

一

二

三

二

一

二

三

二

一

二

三

二

一

二

三

二

一

二

三

二

一

二

三

二

一

二

三

二

一

二

三

二

一

二

三

二

一

二

三

二

一

二

三

二

一

二

三

二

一

二

三

二

一

二

三

二

一

二

三

二

一

二

三

二

一

二

三

二

一

二

三

二

一

二

三

二

一

二

三

二

一

二

三

二

一

二

三

二

一

二

三

二

一

二

三

二

一

二

三

二

一

二

三

二

一

二

三

二

一

二

三

二

一

二

三

二

一

二

三

二

一

二

三

二

一

二

三

二

一

二

三

二

一

二

三

二

一

二

三

二

一

二

三

二

一

二

三

二

一

二

三

二

一

二

三

二

一

二

三

二

一

二

三

二

一

二

三

二

一

二

三

二

一

二

三

二

一

二

三

二

一

二

三

二

一

二

三

二

一

二

三

二

一

二

三

二

一

二

三

二

一

二

三

二

一

二

三

二

一

二

三

二

一

二

三

二

一

二

三

二

一

二

三

二

一

二

三

二

一

二

三

二

一

二

三

二

一

二

三

二

一

二

三

二

一

二

三

二

一

二

三

二

一

二

三

二

一

二

三

二

一

二

三

二

一

二

三

二

一

二

三

二

一

二

三

二

一

二

三

二

一

二

三

二

一

二

三

二

一

二

三

二

一

二

三

二

一

二

三

二

一

二

三

二

一

二

三

二

一

二

三

二

一

二

三

二

一

二

三

二

一

二

三

二

一

二

三

二

一

二

三

二

一

二

三

二

一

二

三

二

一

二

三

二

一

二

三

二

一

二

三

二

一

二

三

二

一

二

三

二

一

二

三

二

一

二

三

二

一

二

三

二

一

二

三

二

一

二

三

二

一

二

三

二

一

二

三

二

一

二

三

二

一

二

三

二

一

二

三

二

一

二

三

二

一

二

三

二

一

二

三

二

一

二

三

二

一

二

三

二

一

二

三

二

一

二

三

二

一

二

三

二

一

二

三

二

一

二

三

二

一

二

三

二

一

二

三

二

一

二

三

二

一

二

三

二

一

二

三

二

一

二

三

二

一

二

三

二

一

二

三

二

一

二

三

二

一

二

三

二

一

二

三

二

一

二

三

二

一

二

三

二

一

二

三

二

一

二

三

二

一

二

三

二

一

二

三

二

一

二

三

二

一

二

三

二

一

二

三

二

一

二

三

二

一

二

三

二

一

二

三

二

一

二

三

二

一

二

三

二

一

二

三

二

一

二

三

二

一

二

三

二

一

二

三

二

一

二

三

二

一

二

三

二

一

二

三

二

一

二

三

二

一

二

三

二

一

二

三

二

一

二

三

二

一

二

三

二

一

二

三

二

一

二

三

二

一

二

三

二

一

二

三

二

一

二

三

二

一

二

三

二

一

二

三

二

一

二

三

二

一

二

三

二

一

二

三

二

一

二

三

二

一

二

三

二

一

二

三

二

一

二

三

二

一

二

三

二

一

二

三

二

一

二

三

二

一

二

三

二

一

二

三

二

一

二

三

二

一

二

三

二

一

二

三

二

一

二

三

二

一

二

三

二

一

二

三

二

一

二

三

二

一

二

三

二

一

二

三

二

一

二

三

二

一

二

三

二

一

二

三

二

一

二

三

二

一

二

三

二

一

二

三

二

一

二

三

二

一

二

三

二

一

二

三

二

一

二

三

二

一

二

三

二

一

二

三

二

一

二

三

二

一

二

三

二

一

二

三

二

一

二

三

二

一

二

三

二

一

二

三

二

一

二

三

二

一

二

三

二

一

二

図：「エージェント別ランキング」と「直近エラー一覧」。行をクリックすると、その条件を引き継いだままトレース詳細へ遷移する。

トレースへの掘り下げ

p90・p95・p99 を超えた外れ値や、特定のエラー種別、「直近エラー一覧」からは、その条件を引き継いだままトレースへ遷移できます。応答時間の外れ値は、いったん概要一覧のダイアログが開き、そこから個別のトレース詳細、またはダイアログ下部のボタンで条件付きのトレース一覧へ進みます。いずれの遷移も別タブで開き、ダッシュボード側の絞り込み状態は失われません。しきい値を設定してアラートを通知する仕組みは用意されていません。

読み違いやすい点

- エラー率は、KPI とランキングではガードレールによるブロックを除外して計算されますが、推移グラフと直近エラーの一覧では、ガードレールによるブロックを含めて計算されます。
- 実行種別は、本番とテストの 2 択で、既定は本番です。テスト実行しか行っていない場合、画面には 0 件と表示されます。
- 単価が未設定のモデルによる実行 1 件のコストは、0 ではなく「- (ハイフン)」と表示されます。一方、「エージェント別ランキング」の「1実行あたりコスト」は複数実行の平均であり、単価未設定の呼び出しを含む行でも分母（実行数）を減らさずに算出します。そのため値は実態より小さく出ますが、値自体は隠さず表示したうえで警告を添えます。
- 直接実行（エージェント定義を経由しない実行）は、呼び出し元を区別せず 1 つのグループへ集約されます。個々の呼び出し元ごとの内訳は見られません。
- エラー種別内訳からトレース一覧へ掘り下げると、検査位置（入力／ツール入力／ツール出力／出力）の区別は引き継がれません。「ガードレール ブロック」で絞り込んだ場合、どの検査位置で止まったかはトレース詳細側で確認します。
- 「経営ダッシュボード」の失敗コストには、ガードレール ブロックで中断した実行も含まれます。ブロックされるまでに LLM を呼び出していれば、その分のトークンは消費されるためです。本章の KPI とランキングのエラー率はガードレール ブロックを除外するため、運用ダッシュボードでエラー率が 0% のエージェントでも、経営ダッシュボードでは失敗コストが計上されることがあります。
- 金額は USD で表示されます。画面右上の「1 USD =」の欄にレートを入力すると、表示だけを別の通貨に換算できます（「コストの見方」）。

経営ダッシュボード

本章の読者は業務管理者・経営層を想定しており、参照のみのロールで到達できます。「[グループ解決設定](#)」だけは管理権限が必要です。

項目

- [何が見える画面か](#)
- [KPI の 5 枚](#)
- [推移と内訳](#)
- [金額を読むときの前提](#)
- [グループ解決設定](#)
- [読み換えやすい点](#)

何が見える画面か

この画面は、LLM の費用と利用状況を集計して、投資判断に活かすためのものです。障害や性能の問いは扱いません（そちらは「[運用ダッシュボード](#)」で扱います）。典型的には、次のような場面で使います。

- 一定期間のコスト・トークン消費を振り返り、前期間と比べてどう動いたかを確認したいとき
- どのエージェント・モデル・部門（グループ）・利用者がコストや利用回数の増加に寄与しているかを特定したいとき
- 単価が未設定のモデルがないか、コストの集計に漏れがないかを点検したいとき

サイトマップから「Copilot」→「Accel Agent」→「ワークスペース」を開き、画面左のサイドナビゲーションの「経営ダッシュボード」から移動します。

集計軸には、エージェント別、モデル別、グループ別、ユーザ別の 4 つがあります。既定の対象期間は 30 日間であり、7 日間を既定とする運用ダッシュボードとは異なります（経営判断を月次の視点で見えるためです）。運用ダッシュボードと名前が同じ指標でも、実行種別（本番／テスト）を含めるかどうかやエラーの数え方が指標ごとに意図的に異なるため、値が一致しないことがあります（「[金額を読むときの前提](#)」）。

KPI の 5 枚

画面上部には、推定総コスト、失敗コスト、総トークン消費、本番実行回数、利用ユーザ数という 5 つの KPI が並びます。いずれも、前の期間との比較つきで表示されます。比較の呼称は期間の長さで変わり、7 日なら前週比、30 日なら前月比、それ以外（90 日、時間単位、カスタム期間）は前期間比を表示します。前期間の実績が無い場合は、比較値の代わりに「-（ハイフン）」が表示されます。

利用ユーザ数は、データの取得に失敗した場合でもエラーが表示されず、「0」と表示されることがあります。推定総コストや本番実行回数に値があるのに利用ユーザ数が「0」になっている場合は、画面を再読み込みしてください。

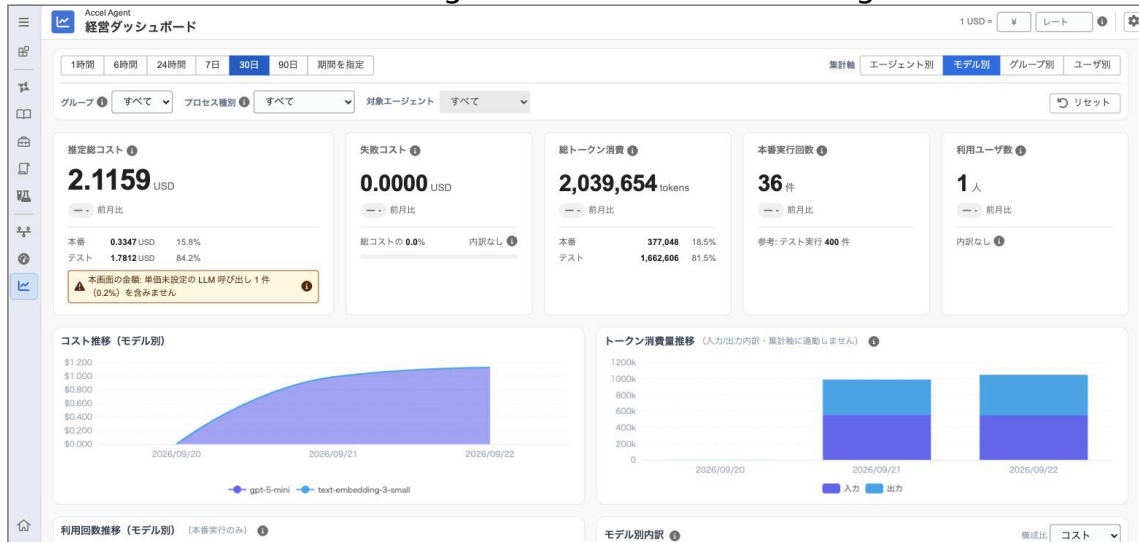


図：経営ダッシュボードの画面上部。絞り込みと集計軸の切り替え、5 つの KPI、コスト・トークン消費量の推移グラフが並ぶ。

推移と内訳

コスト、トークン、利用回数については推移グラフがありますが、利用ユーザ数は KPI として表示されるのみで、推移グラフはありません。トークン消費量の推移だけは集計軸に連動せず、常に入力／出力の 2 系列で固定です（キャッシュ読み込み分は入力トークンの内数のため、積み上げると二重計上になり含めません）。

モデル別の内訳は、集計キーに人が付けた表示名ではなく、記録されたプロバイダのモデル識別子を使います。同じ表示名を複数のモデル定義に付けられるため、表示名からは単価を一意に決められないからです。



図：集計軸を「モデル別」に切り替えた状態。コスト推移のグラフが、モデルごとの系列に切り替わる。

利用統計のランキングは、集計軸を「グループ別」「ユーザ別」にしたときだけその軸に追従し、「エージェント別」「モデル別」を選んだ場合は常にエージェント別で表示されます。並び順は利用回数・コスト・失敗コスト・トークンの列で切り替えられますが、常に降順です。表示件数は「上位10件」「上位30件」「すべて」から選べ、既定は「上位10件」です。上位を選んだ場合、溢れた行は「その他」にまとめられます。

金額を読むときの前提

指標ごとに、実行種別の扱いが異なります。利用回数は本番実行のみを対象とし、コストとトークンは本番・テストの両方を対象とします。

失敗コストは、ステータスがエラーで終了した実行のコスト合計で、母集団は推定総コストと同じ本番・テスト全体です。運用ダッシュボードのエラー率がガードレールブロックを障害として除外するのに対し、失敗コストはガードレールブロック由来のエラーも区別せず含みます。「エラー」という同じ言葉でも、金額の問いと品質の問いでは数えている対象が異なるため、両画面の数字は一致しません。

単価が未設定のモデルが含まれる場合、その旨を示すバッジが表示され、対象のモデルが「プロバイダモデルID（ドライバ名）」の形で注記に列挙されます。この可視化はスパン単位（LLM 呼び出し 1 回ごと）の件数であり、トレース一覧が出す「単価未設定のため金額を算出できない実行 N 件」（そのトレースの LLM 呼び出しがすべて単価未設定だった件数）とは数え方が異なります。1 つの実行の中に単価あり・なしが混在する場合、トレース一覧の件数には現れないまま金額だけが実態より小さくなるため、2 つの数字を突き合わせて一致を期待しないでください。

金額は USD で表示されます。画面右上の「1 USD =」の欄にレートを入力すると、表示だけを別の通貨に換算できます。使い方と注意点は「[コストの見方](#)」を参照してください。

グループ解決設定

グループ解決設定は、経営ダッシュボードの集計軸「グループ別」を使うための設定です。エージェントが実行されるたびに、実行したユーザがどのグループ（部門など）に属するかを判定して記録するため、コストや利用回数を部門ごとに集計できます。設定していない場合、グループ別の集計では、すべての実行が「未分類」として集計されます。

設定は、経営ダッシュボードの歯車アイコンから開く画面で、テナントに 1 件だけ行きます。グループを判定するIM-LogicDesigner のロジックフローと、そのバージョン（未指定の場合は最新バージョン）を指定します。必要な権限は「[参照のみのロールの作り方](#)」を参照してください。

グループを判定するロジックフローは、次の入力値を受け取り、出力値を返すように作成します。

入力値		
項目名	型	説明
<code>user_cd</code>	文字列	エージェントを実行したユーザのユーザコード

出力値		
項目名	型	説明
<code>group_id</code>	文字列	グループID（必須）。集計のキーとして使われます。半角英数字と記号（ <code>_</code> 、 <code>-</code> 、 <code>.</code> 、 <code>@</code> 、 <code>+</code> 、 <code>!</code> ）で、100 文字以内で返します
<code>group_name</code>	文字列	グループ名。画面に表示されます。255 文字以内で返します

！ 注意

グループ解決設定は、過去の実行には遡って適用されません。集計にグループ別の軸を使いたい場合は、本番運用を開始する前に設定しておく必要があります。フローがエラーになる、または想定外の戻り値を返す場合、該当する実行は「未分類」として記録され、実行そのものは継続します。

フローの処理に時間がかかると、その分だけエージェントの実行も遅くなります。判定結果はユーザごとに一定時間キャッシュされるため、遅くなるのは主に各ユーザの初回の実行です。

読み違いやすい点

- キャッシュ効率（HIT率）は、分母（入力トークン）が 0 のとき「0%」ではなく「-（ハイフン）」と表示されます。前期間比を持たない唯一の指標で、並べ替えの対象にもなりません（値を高めるための打ち手が利用者側に無いためです）。ランキングの「その他」行にもこの値は出しません。
- 失敗コストは、実行種別（本番／テスト）ごとの内訳を確認する手段がありません。KPI が返す値は本番・テストを合わせた総額だけです。
- グループ・対象エージェントの選択肢は、期間だけで決まります。プロセス種別やモデルなど他の絞り込みをかけても、選択肢の一覧は変化しません。

監査ログ

本章の読者は監査担当を想定しています。参照のみのロールで到達できますが、画面はサイドナビゲーションには無く、サイトマップからのみ開きます。

項目

- [何が記録されるか](#)
- [記録されないもの](#)
- [変更内容の粒度](#)
- [検索と CSV](#)
- [トレースとの行き来](#)
- [保持と増加](#)

何が記録されるか

この画面は、管理操作とランタイム事象の記録を、期間・操作種別・リソース種別・操作者で絞り込んで選んで読むためのものです。トレースが「1回の実行の中身」を見る画面であるのに対し、監査ログは「設定を変えた操作」と「ガードレールが発動した事象」を横断して見る画面です。典型的には、次のような場面で使います。

- 設定が意図した通りに変更されているか、誰がいつ変更したかを確認したいとき
- ガードレールがいつ・どの設定で発動したかを、証拠として確認したいとき
- 外部への提出・長期保管のため、変更履歴を CSV で保存しておきたいとき

画面は左に一覧、右に詳細ペインを置く 2 ペイン構成です。一覧の行を選ばると、右のペインにその行の全項目が表示されます。

The screenshot displays the 'Accel Agent 監査ログ' (Audit Log) interface. At the top, it shows '28 ログ件数' (28 log entries) and '0 ガードレール ブロック' (0 guardrail blocks). Below this are filters for time period (1時間, 6時間, 24時間, 7日, 30日, 90日, 期間を指定), operation type (すべて), resource type (ナレッジベース), and user code. A 'CSV エクスポート' button is also present. The main table lists log entries with columns: 日時 (Date/Time), ユーザ (User), 操作種別 (Operation Type), リソース種別 (Resource Type), リソース名 (Resource Name), and 結果 (Result). The right-hand panel provides a detailed view of the selected entry, showing the '購買承認規定' (Purchase Approval Regulation) and the specific update performed: 'UPDATE: description'.

図：監査ログ画面。検索条件（期間・操作種別・リソース種別・ユーザコード）、一覧、選択した行の詳細ペインが並ぶ。

記録の対象には、エージェント定義、ロジックツール、MCPサーバ、ナレッジベース、スキル、AIモデル、ガードレール、評価計画などがあります。操作種別には、作成・編集・削除・公開などの操作のほか、ガードレール発動やベクトル化完了もあります。「無効化」は、バージョンの公開を取り下げる操作を指します。

記録されないもの

ガードレールは、ブロックした場合にだけ記録されます。「記録する」設定による検知結果は、監査ログには残りません。件数を確認したい場合は、トレースを見ます。ガードレールが検知した原文や、判定の文言そのものも、監査ログには残りません。

変更内容の粒度

変更内容は、1本の要約文として記録されます。自由記述の項目や秘匿情報については、項目名だけが記録され、値そのものは記録されません。そのため、「何が変わったか」は分かっても、「何から何に変わったか」までは分からない項目があります。

検索と CSV

検索条件には、期間（1時間・6時間・24時間・7日・30日・90日・「期間を指定」から選択、既定は30日）、ユーザコード（完全一致）、操作種別、リソース種別があります。ユーザコードは完全一致で検索するため、一部だけを入力しても該当しません。並び替えは、日時（既定は降順）、ユーザ、操作種別の3列に限られます。1ページは20件です。

CSV エクスポートには 10 万行の上限があり、超えた場合はエクスポート自体が拒否されます（切り詰めての出力はしません）。出力は UTF-8（BOM 付き）で、対象範囲は表示中のページではなく検索条件に一致する全件です。出力される値はコード値のままであり、画面の表示語には変換されません。可観測性の画面の中で、エクスポート機能を持つのはこの監査ログだけです。

トレースとの行き来

ランタイム（エージェントの実行）に由来する行には、関連するトレースへのリンクが付きます。管理操作に由来する行には、このリンクは付きません。遷移は別タブで開き、監査ログ側の絞り込み状態は失われません。

保持と増加

監査ログの保持期間の設定やパージの機能は、初版では提供されません。そのため、記録は消えることなく増え続けます。定義体の変更は、監査ログとは別に、intra-mart Accel Platform 標準のマスタデータ更新ログにも出力されます。

ジョブの運用

本章の読者はテナント管理者を想定しています。テナント環境セットアップの直後は、「[ジョブネットの作成と有効化](#)」の作業を済ませれば、ナレッジベースを使い始められます。

項目

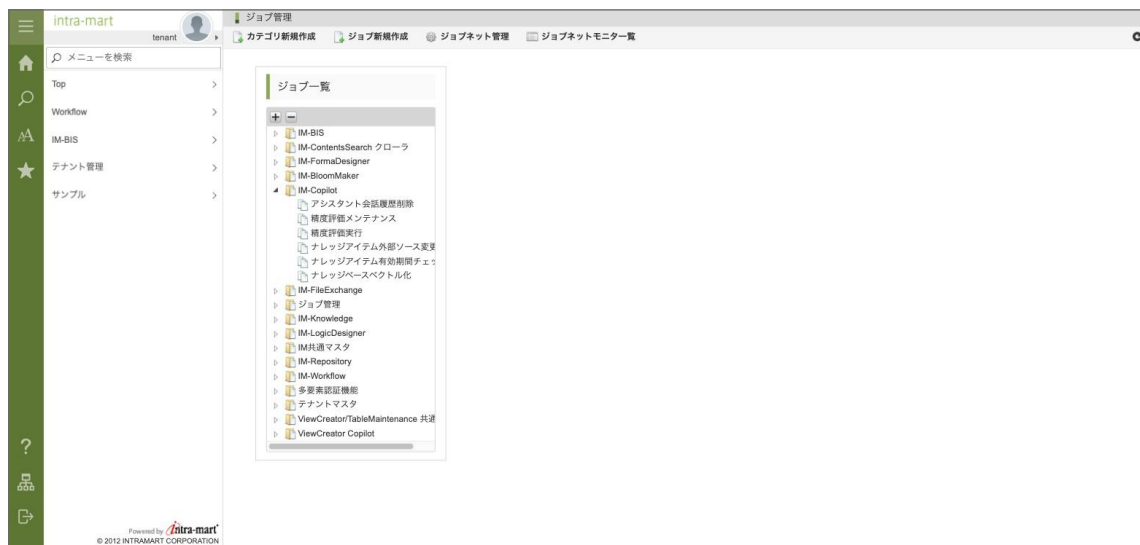
- [ジョブネットの作成と有効化](#)
- [ジョブごとの設定](#)
- [運用の注意](#)

ジョブネットの作成と有効化

カテゴリ「IM-Copilot」には、Accel Agent 用の次の 5 つのジョブがあります。

ジョブ	役割	動いていない場合
ナレッジベースベクトル化	「待機中」のナレッジアイテムをベクトル化し、検索できる状態にする	登録・更新したアイテムが「待機中」のまま、検索されない
ナレッジアイテム有効期間チェック	有効期間の開始日を迎えたアイテムをベクトル化の対象（「待機中」）に戻し、終了日を過ぎたアイテムを検索対象から外す	有効期間を設定したアイテムが、期間どおりに検索対象へ入ったり外れたりしない
ナレッジアイテム外部ソース変更検知	外部ソース変更検知を有効にした IM-Wiki コンテンツの更新を検知し、そのアイテムを再ベクトル化の対象に戻す	IM-Wiki 側の更新が、ナレッジベースに反映されない
精度評価実行	評価計画を指定して、精度評価を定期的に実行する	定期的な評価が行われない（画面から開始する評価には影響しない）
精度評価メンテナンス	保持期間を過ぎた評価の詳細データと生成記録を削除し、実行中・生成中のまま止まった処理をエラーにする	詳細データが削除されずにたまり続け、止まった評価が「実行中」のまま残る

ナレッジベースを使う場合は、上の 3 つのジョブが必要です。精度評価の 2 つのジョブは、定期的な評価やデータの整理を行う場合に設定します。これらのジョブネットは製品に同梱されないため、管理者が作成して有効化する必要があります。手順は テナント管理者操作ガイドの[ジョブを設定する](#)を参照してください。



図：ジョブ管理画面のカテゴリ「IM-Copilot」。ナレッジベースベクトル化などのジョブが並ぶ。ジョブネットの作成・有効化は、この一覧に対して管理者が行う作業である。

コラム

同じカテゴリには、8.0.4 由来の「アシスタント会話履歴削除」というジョブも並んでいます。これはIM-Copilotのアシスタント機能の会話履歴を削除するジョブで、Accel Agent の動作には関係しません。そのため、画面上は合計 6 つのジョブが見えます。

ジョブごとの設定

どのジョブも、実行周期はジョブネットのトリガで設定します。実行時のパラメータを持つのは精度評価に関する 2 つのジョブだけで、ナレッジベースに関する 3 つのジョブはパラメータを持ちません。

精度評価実行のジョブは、評価計画を全件で実行します。先頭の数件だけを実行する少数実行は、ジョブからは指定できません。評価計画ID 以外のパラメータは任意で、指定しなかった項目には評価計画の設定値が使われます。数値のパラメータに数値として解釈できない値を指定した場合は、エラーで止まります。

精度評価実行のパラメータ

キー	必須	説明
planId	○	実行する評価計画の ID

キー	必須	説明
agentVersion	—	評価対象エージェントのバージョン番号。省略時は、非公開を含めて取得できる最大のバージョン番号を使う
judgeModelId	—	採点モデルの AIモデルID
embeddingModelId	—	Embeddingsモデルの AIモデルID
promptVersion	—	採点に使う判定プロンプトのバージョン。現在のバージョンは v1.0 だけのため、通常は指定しない
judgeSamples	—	採点回数（正の整数）

精度評価メンテナンスのパラメータ

キー	既定値	説明
retentionDays	180	保持日数。作成からこの日数を過ぎた評価の詳細データと生成記録を削除する
staleMinutes	120	実行中・生成中のまま止まっているとみなすまでの時間（分）。この時間を過ぎた評価実行と自動生成をエラーにする

運用の注意

精度評価メンテナンスのジョブネットを作成してトリガを設定するまで、評価の詳細データと生成記録は自動では削除されません。このジョブは、保持期間を過ぎたデータを 1 回の処理でまとめて削除します。長期間動かしていなかった環境で初めて実行する場合は、`retentionDays`を大きめの値にして実行し、段階的に短くしていきます。

精度評価メンテナンスのジョブに不正なパラメータ（数値として解釈できない値や 1 未満の値）を指定した場合は、既定値に読み替えられず、削除を始める前にエラーで止まります。

ロールと権限

Accel Agent には、管理作業に必要な権限をまとめた「Accel Agent 管理者」ロールを同梱しています。本章では、同梱ロールの権限と、参照のみのロールを作るときの認可設定を説明します。本章の読者はテナント管理者を想定しています。

項目

- 同梱ロールが許すもの
- 参照のみのロールの作り方
 - 役割ごとの設定例
- 参照だけでは足りない場面
- 権限と画面の見え方
- 見える範囲の広さ

同梱ロールが許すもの

「Accel Agent 管理者」ロールは、Accel Agent の各機能に加えて、ロジックフローの編集と OAuth プロバイダ設定も許可します。「LogicDesigner 管理者」ロールには、ロジックフローのタスクでエージェントを選べるよう、エージェント定義の参照権限が付与されています。

ナレッジベースのアイテム追加で IM-Wiki コンテンツを選ぶには、IM-Knowledge の認可リソース「Knowledge コンテンツ利用者」の実行権限が必要です。選択には IM-Knowledge のコンテンツ検索画面を使うためです。「Accel Agent 管理者」ロールにはこの権限が含まれていないため、次のいずれかを設定します。

- 認可設定画面で、認可リソース「Knowledge コンテンツ利用者」の実行を「Accel Agent 管理者」ロールに許可する
- 操作するユーザに、ロール「Knowledge コンテンツ利用者」を付与する

ロール「テナント管理者」を持つユーザには、標準でこの権限が許可されているため、設定は不要です。

参照のみのロールの作り方

参照のみのロールは、ロールを作成したうえで、認可設定画面で必要なリソースだけを許可して作ります。主な認可リソースは、リソースグループ「IM-Copilot」の配下にあり、機能ごとに参照と管理に分かれています。テスト実行の認可は、別のリソースグループにあります。

Accel Agent の認可リソース

リソースグループ	リソース	許可すると使えるもの
Accel Agent	エージェント (参照)	ワークスペース、エージェント一覧の表示。IM-LogicDesignerの「エージェント実行」タスクでのエージェントの選択
	エージェント (管理)	エージェント定義の作成・編集・削除、バージョンの公開・非公開、インポート・エクスポート
	ツール (参照)	エージェント定義でツールを選ぶための候補の取得（この権限だけで開ける画面はありません）
	ツール (管理)	ツール管理画面、ツールのインポート・エクスポート
	ナレッジベース (参照)	エージェント定義でナレッジベースを選ぶための候補の取得（この権限だけで開ける画面はありません）
	ナレッジベース (管理)	ナレッジベース管理画面、ナレッジベースのインポート・エクスポート
	スキル (参照)	エージェント定義でスキルを選ぶための候補の取得（この権限だけで開ける画面はありません）
	スキル (管理)	スキル管理画面、スキルのインポート・エクスポート
	精度評価 (参照)	評価の計画・結果・推移・レポートの表示
	精度評価 (管理)	評価計画の作成、データセット画面の表示・取り込み・編集、評価の実行・中断・再開・再採点、改善提案
AIモデル	AIモデル (参照)	他の画面で AIモデルを選ぶための候補の取得（この権限だけで開ける画面はありません）
	AIモデル (管理)	AIモデル管理画面、AIモデルのインポート・エクスポート
可観測性	トレースビューア (参照)	トレース一覧・詳細
	メトリクスダッシュボード (参照)	運用ダッシュボード、経営ダッシュボード
	監査ログビューア (参照)	監査ログの表示と CSV エクスポート
	グループ解決設定 (参照)	経営ダッシュボードのグループ解決設定の表示
	グループ解決設定 (管理)	グループ解決設定の保存・削除
IM-Copilot アシスタントAPI	Accel Agent テスト実行アシスタント	エージェント編集画面でのテスト実行

役割ごとの設定例

「[想定読者と必要な権限](#)」で挙げた読者のうち、参照のみのロールで足りる読者には、次のリソースを許可します。

役割ごとに許可するリソースの例

役割	許可するリソース	補足
運用管理者	トレースビューア (参照)、メトリクスダッシュボード (参照)、エージェント (参照)	運用ダッシュボードは、直近のエラーなどをトレースから読み込むため、トレースビューア (参照)も必要です。エージェント (参照)は、トレース一覧でバージョンを絞り込むときにも使います。
業務管理者・経営層	メトリクスダッシュボード (参照)、エージェント (参照)	経営ダッシュボードを参照するだけであれば、これで足りります。
監査担当	監査ログビューア (参照)	監査ログは、サイトマップからのみ開けます（「 権限と画面の見え方 」）。

いずれの場合も、サイトマップに Accel Agent のメニューを表示するため、メニューグループ「Copilot」の参照を許可します。

運用ダッシュボード、経営ダッシュボード、トレースは、サイトマップのメニューに含まれていません。これらの画面へは、サイトマップの「ワークスペース」を開き、画面左のサイドナビゲーションから移動します。ワークスペースを開くにはエージェント (参照)が必要なため、運用管理者と業務管理者・経営層にもエージェント (参照)を許可します。エージェント (参照)を許可すると、ダッシュボードやトレースだけでなく、エージェント一覧の画面も開けます。



コラム

グループ解決設定を保存するには、グループ解決設定 (参照)・(管理)に加えて、IM-LogicDesignerの認可リソース「IM-LogicDesigner 管理」も必要です。設定画面では、ロジックフローをIM-LogicDesignerの一覧から選ぶためです。「IM-LogicDesigner 管理」はロジックフローの編集も許可するため、グループ解決設定は、これらの権限をすでに持つ Accel Agent 管理者ロールの利用者が行う運用を推奨します。

参照だけでは足りない場面

ツール管理、ナレッジベース管理、スキル管理、AIモデル管理の各画面は、管理権限で保護されています。参照権限だけで到達できるのは、ワークスペース、エージェント一覧、精度評価の参照系画面、トレース、ダッシュボード、監査ログに限られます。定義のインポート・エクスポート（「[インポート／エクスポート](#)」）にも、管理権限が必要です。

精度評価にも、管理権限が必要な画面があります。参照権限では計画・結果・推移・レポートを確認できます。データセット画面を開くには管理権限が必要です。改善提案と添付ファイルの参照にも管理権限が要ります。

権限と画面の見え方

サイドナビゲーションの項目は、認可によってフィルタされません。そのため、権限を持たない項目もメニューには表示され、実際に開こうとするとエラーが発生します。監査ログは、サイドナビゲーションには表示されず、サイトマップからのみ到達できます。

見える範囲の広さ

トレースと監査ログには、行単位のアクセス制御がありません。参照権限を許可された利用者は、テナント内の全件を見ることができます。エージェント単位で閲覧を制限する仕組みはありません。

