

Clearer work, better evidence

Find the right person behind a record

Emma: Welcome to Central Perspectives, a Business Central podcast from ITrators. I'm Emma, here with Alex. We're AI hosts, and our scripts and voices are AI-generated. Today, two changes in Business Central twenty-nine make information easier to investigate: audit details on your lists, and AI-assisted capture when something fails. Alongside those October changes, we have three useful developments from earlier this year, including different sales agents for different incoming work. The question running through them is simple: what can the person doing the work now see or decide more clearly? Now, let's get started.

[Run multiple Sales Order Agents per company](#)

[Create purchase quotes for contacts](#)

[Define item attributes for item variants](#)

[Update 29.0 for Business Central 2026 release wave 2](#)

[What's New or Changed in Business Central 2026 release wave 2 - Update 29.0 - Business Central](#)

Emma: Let's begin with a small October change that could save a surprisingly vague conversation. Business Central twenty-nine lets you use system audit fields in analysis tabs, and add them to pages through profiles. So information about who created or last modified a record becomes easier to bring into view. Alex, what is the useful question this answers, before we turn it into a grand auditing project?

[Update 29.0 for Business Central 2026 release wave 2](#)

[Feature details for Business Central 2026 release wave 2](#)

Alex: Often it's simply, who should I ask about this record? Imagine a team checking a list before handing work to the next person. Something looks different from yesterday. The last-modified time and user can give them a starting point without asking everybody in the room. The useful change is easier access to that information. It isn't that Business Central has only just begun storing creation and modification details; those fields already existed.

[Feature details for Business Central 2026 release wave 2](#)

[Work with table system fields](#)

Emma: A starting point sounds more useful than a culprit. If the name says Alex, I can already imagine somebody announcing that Alex changed the price. But the record might have been changed again for a completely different reason. How much can I safely read into that last-modified label?

[Work with table system fields](#)

Alex: Only that it was the last modification to the record, not that Alex changed that particular price. These fields don't contain all the intervening changes or the old and new values of each field. So the name helps you find a useful starting point. It doesn't supply the missing history.

[Work with table system fields](#)

Emma: Then I'd rather ask, which records have been touched since we last checked them? That doesn't accuse anyone, and it helps me choose where to look. Could a team use the new analysis access for that sort of first pass, without changing the records while they explore?

[Feature details for Business Central 2026 release wave 2](#)

[Analyze list page and query data using data analysis](#)

Alex: Yes. Analysis doesn't edit the underlying records. Imagine the team has a hundred records to review, but the useful question is which ones were modified after yesterday's handover. A last-modified date gives that investigation a much better starting point than opening all hundred. Then the team can inspect the relevant records and ask a focused question. They do need analysis permission, and individual pages can have analysis disabled, so start with a page they can actually use.

[Feature details for Business Central 2026 release wave 2](#)

[Work with table system fields](#)

[Analyze list page and query data using data analysis](#)

Emma: That changes the handover conversation. Instead of everyone trying to remember what they touched, we have a smaller set of records in front of us. I'd still want to distinguish something newly created from an existing record somebody updated. Otherwise a busy morning of new work could look like a morning spent correcting old work.

[Work with table system fields](#)

Alex: And that's where the separate creation and modification information helps. Creation details stay with the record; modification details change when it is updated. So your team can ask two different questions: what arrived, and what was subsequently touched? Neither number is a productivity score. But together they can guide a review of the work, instead of leaving somebody to reconstruct the morning from memory.

[Work with table system fields](#)

Emma: Now the profile option makes more sense to me. If that question keeps coming up, an administrator can bring the relevant fields into the team's page, instead of everyone discovering the same useful view separately. Not more information for its own sake. Less time spent finding the information that starts the conversation.

[Feature details for Business Central 2026 release wave 2](#)

Alex: That's the payoff I'd look for: can the next person pick up the work more easily? Try the view with an actual handover question, and keep the fields that help answer it. This is a fairly small release item, but it can sit right inside a task people already do, rather than asking them to learn a whole new process.

[Feature details for Business Central 2026 release wave 2](#)

Give incoming sales work its own rules

Emma: Now let's move from seeing who handled work to deciding where incoming work should go. Business Central gained support for multiple Sales Order Agent instances in one company in July, not this October. Each instance can have its own email monitoring and processing rules, with separate metrics. That sounds more interesting to me than simply having more agents. Why would a company actually want more than one?

[Run multiple Sales Order Agents per company](#)

[Set up Sales Order Agent](#)

Alex: Because different incoming requests can need different next steps, even inside one company. One group of customers may want to see and confirm a quote before an order is created. Another stream may need quotes prepared for the sales team to handle. Separate instances let you give those streams their own processing rules and see their metrics independently. The benefit isn't simply doing more at once; it's making the setup fit the kind of work coming in.

[Set up Sales Order Agent](#)

Emma: I like the distinction between preparing a quote and deciding that an order should follow. Suppose one stream is fairly routine, while the other often leads to a price discussion. We can organize them differently without expecting the second agent to become an expert negotiator, can we?

[Set up Sales Order Agent](#)

[FAQ for Sales Order Agent](#)

Alex: Exactly. Price and discount changes still need people; another instance doesn't remove that limit. But separating supported routine work from commercial decisions can be valuable in itself. I'd keep those negotiations with a person, and configure the agent-supported path around the quote and order decisions it actually supports. Then the sales team knows what kind of result to expect, rather than checking every task against a vague promise of automation.

[Set up Sales Order Agent](#)

[FAQ for Sales Order Agent](#)

Emma: Then let's make the quote decision precise. I can imagine saying, send this customer a quote, but don't make the order yet. Or keep the quote inside the company for someone to finish. Are those separate choices in the setup, rather than one big automation switch?

[Set up Sales Order Agent](#)

Alex: Take what happens to the quote. The agent starts by creating a sales quote, but sending that quote to the customer for confirmation and converting it to an order are separate settings. One stream could send the quote and wait for confirmation before conversion. Another could leave the quote for a person to convert. There is also a configuration for conversion without sending the quote first. Those are different commercial processes, not just different speeds.

[Set up Sales Order Agent](#)

Emma: I'd be cautious about calling that last one the advanced setting. Waiting for a customer to confirm can be the sensible process, not a sign that the company hasn't automated enough. If the customer is still comparing options, a fast order might be exactly the wrong outcome.

[Set up Sales Order Agent](#)

Alex: Absolutely. I'd choose the setting from the agreement with the customer, not from how autonomous it sounds. And there are two different confirmations people can accidentally collapse into one. Customer confirmation of a quote is one thing; the configured human review of documents is another. Turning off the request for customer confirmation doesn't, by itself, mean all human review vanishes. Before changing a stream, draw the intended path from enquiry to quote to order, and identify which decision belongs to whom.

[Set up Sales Order Agent](#)

Emma: And the incoming stream itself has to mean something. Microsoft now documents monitoring a selected folder in a Microsoft three-sixty-five user or shared mailbox. I can see why a team might prefer a clearly separated folder to feeding every message in the company's main inbox to an agent. But that still leaves somebody deciding which work belongs there.

[Set up Sales Order Agent](#)

Alex: Yes, the folder is a boundary you design; it isn't a substitute for deciding what belongs in the stream. And you don't have to connect a mailbox to explore the workflow at all. Microsoft also supports starting a task manually. That gives the team a way to consider a known request and the expected result before introducing continuous email intake. It's still an agent task, though, with the relevant access and consumption billing, not a free simulation.

[Set up Sales Order Agent](#)

[Manage consumption-based billing for agent capabilities](#)

Emma: That seems a more approachable first step than reorganizing the whole inbox. Pick a request whose intended outcome we understand, and see whether the resulting quote actually gets us there. What would you look at when deciding whether to add the second instance? Separate metrics sound helpful, but I don't want the busiest agent to win just because it created the most activity.

[Set up Sales Order Agent](#)

Alex: I'd compare outcomes for the kinds of requests in each stream, alongside the team's own review of the resulting documents. Was the quote useful? Did the chosen confirmation path suit the customer? How much human follow-up remained? Those are evaluation questions, not promises about built-in dashboard fields. Separate instance metrics help keep the streams distinguishable, while Copilot Credit consumption adds the cost side. More generated activity isn't necessarily more useful work, especially if somebody then has to untangle it.

[Set up Sales Order Agent](#)

[Manage consumption-based billing for agent capabilities](#)

Emma: There is one very practical loose end in the documentation: stopping a task doesn't necessarily clear away the quote it already created. That's worth knowing before a team starts experimenting with different streams. Otherwise stop can sound like a reassuring undo button when it isn't.

[Process sales quotes and orders with Sales Order Agent](#)

Alex: It isn't undo, and a stopped task can't be restarted. A quote can remain open and need manual follow-up. So include that result when the team tries the process: not just the smooth request, but a request they deliberately stop and then inspect. The agent also doesn't post documents. Its work sits within the sales process, with later business steps still to complete. That helps define a sensible finish line for the agent rather than treating an order as the whole transaction.

[FAQ for Sales Order Agent](#)

[Process sales quotes and orders with Sales Order Agent](#)

Emma: Before we call a stream routine, there's another question: routine for whom? A familiar-looking email could come from someone who isn't yet a customer in Business Central, or ask for an item that hasn't been created. Does the agent create that missing information as part of processing the request?

[FAQ for Sales Order Agent](#)

Alex: No. The documented agent works with existing customers, contacts and items; it doesn't create those missing records. It supports item-type sales lines, with up to fifteen item lines in the documented scope. That gives us a more concrete definition of the first stream to try: requests for known items from known business relationships, within the supported document shape. A request can be easy for an experienced employee precisely because they know how to fill in missing information. That doesn't mean the same request is ready for the agent.

[FAQ for Sales Order Agent](#)

Emma: That's a much fairer starting point than asking it to handle the whole inbox and judging it on whatever arrives. I'd choose a handful of representative requests from that known stream, including one that should remain a quote and one intended to become an order. Then the team can compare the actual result with the decision they meant to make, not simply celebrate that something happened.

[Set up Sales Order Agent](#)

[FAQ for Sales Order Agent](#)

Emma: For listeners thinking this sounds relevant, Danish is among the supported and validated languages. But language isn't the only setup question. There are permissions for configuring agents, mailbox access where email is involved, and billing to arrange. Is there anything about availability that people should check before assuming this is ready in their particular environment?

[Set up Sales Order Agent](#)

[FAQ for Sales Order Agent](#)

Alex: Check the environment's availability and data-movement settings as well. Where the required Azure OpenAI service isn't available locally, the documented setup requires administrative permission for cross-geography processing. That's a real decision for the business, not something a language setting resolves. With those prerequisites in place, I'd still begin with one clearly described stream. Add another when its rules or incoming work are meaningfully different. Multiple instances are useful because they let you preserve those differences, not because a company ought to collect agents.

[Set up Sales Order Agent](#)

[FAQ for Sales Order Agent](#)

[Manage consumption-based billing for agent capabilities](#)

Make product differences easier to see

Emma: Put the agents aside for a moment. I want to look at something the person choosing a product can actually see. Since April, Business Central has supported attributes on individual item variants, and pictures for those variants too. Not just one description and image for the parent item. Alex, when does that become more than a nicer-looking product card?

[Add pictures to item variants to differentiate product options](#)

[Define item attributes for item variants](#)

Alex: When the difference between variants matters to choosing the right one. Imagine a company selling the same design in two finishes. The item-level description explains what the product is, but the finish and picture tell the person which version they're looking at. A variant code can identify the choice precisely; an attribute and image can make that choice understandable. That's especially useful when two codes are perfectly clear to the system and not particularly memorable to the person.

[Add pictures to item variants to differentiate product options](#)

[Define item attributes for item variants](#)

Emma: The code can be correct and still make someone stop and think. I'm picturing two finishes with almost identical names, where a small image would settle the question immediately. But if we have to enter every shared attribute again for every variant, the person maintaining the information gets a rather less attractive improvement.

[Add pictures to item variants to differentiate product options](#)

[Define item attributes for item variants](#)

[Manage product variants](#)

Alex: Fortunately, a new variant receives attributes from the item. You start with the common information, then change or remove values where the variant needs something different. So in our example, the shared design information can come from the item, while the finish describes the specific option. The interesting work is deciding which information is genuinely common. If everything is treated as a special case, inheritance doesn't save much. If nothing can differ, the variant information doesn't help the person choose.

[Define item attributes for item variants](#)

[Manage product variants](#)

Emma: Let me push on the word receives. We add the variant today, then next month someone corrects an attribute on the main item. Does receives mean the variants stay in step forever, or have we just described the starting point? That's the sort of word that can quietly turn into an assumption.

[Define item attributes for item variants](#)

[Manage product variants](#)

Alex: We've described the starting point, not a promise of continuous synchronization. There is a deliberate Update Variant Attributes action on the item card for forcing a synchronization. Before using that across a product with intentional differences, I would check the outcome on representative variants. In particular, include one that still follows the item and one whose value has deliberately been changed. The documentation gives us the action; it doesn't justify guessing every overwrite consequence for a particular setup.

[Define item attributes for item variants](#)

[Manage product variants](#)

Emma: Then I'd put the intentional difference in the test, not just the easy variant that matches everything. Otherwise we could congratulate ourselves on updating the common values and miss the very detail that made the variant useful. For our two finishes, the job is to keep the shared description accurate without making the choices indistinguishable again.

[Define item attributes for item variants](#)

Alex: And keep the picture just as purposeful. The variant card lets you import a picture, take one with a camera, export it or delete it. Tile views can then make similar choices easier to distinguish. I'd choose images that show the actual difference: the finish, the shape, whatever the person needs to recognize. A beautiful overall product image repeated on every variant can be less useful than a plain image that makes the distinction obvious.

[Add pictures to item variants to differentiate product options](#)

Emma: That's refreshingly unglamorous. The best picture is the one that stops me choosing the wrong finish, not necessarily the one for the front of the brochure. Would you use this as a reason to reorganize a whole item catalogue into variants, or is that getting carried away with a useful improvement?

[Add pictures to item variants to differentiate product options](#)

[Define item attributes for item variants](#)

Alex: I'd begin with products already sensibly represented as variants. This change improves their descriptive information; it doesn't decide your entire item structure. Pick a product where people genuinely hesitate between options, maintain the meaningful attributes and pictures, and ask whether the choice is clearer. That gives you a reason to expand the approach. It avoids turning an April usability improvement into a catalogue redesign before you've learned whether it solves the original problem.

[Add pictures to item variants to differentiate product options](#)

[Define item attributes for item variants](#)

Emma: There is also an existing setting for requiring a variant when variants exist. That isn't part of the new April announcement, but it seems relevant to our example. Clear pictures help a person choose. They don't necessarily make somebody choose at all.

[Manage product variants](#)

Alex: Right. That existing configuration can require a variant for posting when the item has variants, with an item-level override. It's a different job from the picture and attributes. One helps make the selection understandable; the other can require the selection in the relevant posting flow. If the company already uses variants, those two questions belong together: can people recognize the right option, and where must the option be specified? You don't need to pretend they're the same feature to make them useful together.

[Add pictures to item variants to differentiate product options](#)

[Define item attributes for item variants](#)

[Manage product variants](#)

Ask for a quote before creating the vendor

Emma: Here's another earlier-this-year improvement, this time on the buying side. Since April, you can create a purchase quote for a contact that isn't yet linked to a vendor. That lets a company ask a potential supplier for a price before creating the vendor record. It sounds wonderfully straightforward. Alex, where does that temporary, exploratory stage actually end?

[Create purchase quotes for contacts](#)

[Request quotes](#)

Alex: At either of two actions: releasing the quote, or converting it into an order. Either action creates the vendor using a template. So while you're preparing the enquiry, the contact can remain just a contact. But Release isn't merely tidying the status of that quote; it crosses the vendor-creation boundary. That's the important detail to understand before telling a purchasing team that vendor setup always waits until they've decided to order.

[Request quotes](#)

Emma: That's earlier than the neat headline suggests. Suppose I'm comparing a possible new supplier with one we already use. The useful part is being able to begin the enquiry without turning every possibility into a vendor. But if our normal habit is to release quotes straightaway, that habit affects how much of the benefit we actually get.

[Request quotes](#)

Alex: Exactly, so look at the actual enquiry process rather than just the new field choice. Where does the purchaser need a draft? When do they release it, and why? The improvement removes the need for a vendor at the very beginning, but the team should know when the system will create one. There's still a useful saving if a large number of possible suppliers never get that far. And when one does progress, the template can provide a consistent starting point for the vendor.

[Request quotes](#)

Emma: The template is doing real work, then. Does the purchaser choose it every time, or can Business Central work out which one applies? I'm thinking of the person who wants to ask about a delivery date, not suddenly become the keeper of vendor configuration.

[Request quotes](#)

Alex: The matching uses contact criteria including territory, country or region, and contact type. If one template matches, it can be applied automatically; otherwise the user is asked to choose. That's a reason to make the available choices understandable and review the templates before rolling the process out. A clear match can spare the purchaser a decision they shouldn't have to improvise. But creating the vendor this way doesn't prove that the supplier has passed the company's approval process.

[Request quotes](#)

Emma: So the purchaser's first question can remain, is this a supplier we might buy from? The setup should support that question, rather than force a complete vendor decision before an enquiry exists. And once we do decide to buy, what happens to the quote itself? I'd want the order to retain the connection, not leave us guessing which conversation produced it.

[Request quotes](#)

Alex: Make Order creates the purchase order and removes the quote, while the Quote Number carries the originating number onto the order and invoice. So there is a documented connection, but don't expect the quote to remain as an unchanged open working document beside the order. Also, this isn't a direct quote-to-purchase-invoice conversion. If the company wants a different purchasing route, check that route separately rather than assuming every next document is interchangeable.

[Request quotes](#)

Emma: I would start by trying one prospective contact that fits a template and one where the choice isn't obvious. That would show the purchaser both paths before they're in a hurry. It's a small change with a very recognizable benefit: you can begin asking a buying question without first pretending the relationship is further along than it is.

[Request quotes](#)

Capture the failure while it happens

Emma: We've spent most of this conversation on people carrying work forward. What happens when the work stops with an error? Our final October change is AI-assisted snapshot capture in Business Central twenty-nine. You can describe a failing scenario, have an agent prepare the recording, reproduce the problem, and collect the snapshot for investigation. Alex, what is new here, and why is a snapshot more useful than another screenshot of the error?

[Snapshot debugging - Business Central](#)

[What's New or Changed in Business Central 2026 release wave 2 - Update 29.0 - Business Central](#)

Alex: The new part is the AI-assisted capture workflow, not the invention of snapshot debugging. The underlying recording mechanism has been available for years. A screenshot shows what the user saw; a snapshot can preserve recorded code execution and selected variable information for later inspection. That gives a developer something closer to what actually happened inside the system during the reproduction. And the recording doesn't need to stop the user's execution at each point the way a conventional debugging breakpoint would.

[Snapshot debugging - Business Central](#)

Emma: So if somebody says, it failed yesterday, this doesn't magically recover yesterday's missing recording. We need another occurrence with capture ready. I like the prospect of a less awkward handover, but let's keep the user in the picture. What would they actually be asked to do in a sensible support conversation?

[Snapshot debugging - Business Central](#)

Alex: First, agree on a specific failing action. For a hypothetical posting error, that might mean the relevant document and the action that produces the message. The authorized person prepares capture for the appropriate session or next matching reproduction. Once it's armed, the user repeats that agreed action, and the recording is stopped and collected afterward. The agent helps coordinate those capture steps. It doesn't remove the need to know which failure you're trying to reproduce or whether repeating that business action is appropriate.

[Snapshot debugging - Business Central](#)

Emma: That last point keeps it from sounding like, just click Post again until the computer tells us something interesting. The person still needs to understand what they're repeating. And the support person needs to be ready at the right time. Can they leave this armed all afternoon while the user gets around to it?

[Snapshot debugging - Business Central](#)

Alex: No, the waiting capture expires if it hasn't attached within thirty minutes. Once recording starts, it must finish within ten minutes. Those bounds favor a short, coordinated reproduction rather than leaving capture running as background surveillance. In our example, I'd have the user ready on the relevant document, establish that capture is waiting, reproduce the agreed problem, and collect the result. The status matters: waiting, recording and having a downloaded snapshot are different stages, not three names for success.

[Snapshot debugging - Business Central](#)

Emma: Waiting, recording, collected. That's a helpful sequence for somebody who's not the developer as well: the request to capture isn't the evidence yet. Once the file is collected, can the developer investigate it later without keeping the user sitting there through every step of the investigation?

[Snapshot debugging - Business Central](#)

Alex: That's one of the useful benefits. The recorded snapshot can be opened for offline investigation in Visual Studio Code. The developer can inspect the captured execution rather than asking the user to reproduce the same thing while every possible explanation is discussed. Of course, a recording may reveal that another, more focused capture is needed. But the first handover can contain evidence from the failure, not just a description of it. That makes the next question more informed.

[Snapshot debugging - Business Central](#)

Emma: You said selected variable information earlier. If the capture has a Full option, wouldn't it be tempting to choose that every time and avoid having to ask the user twice? More information sounds like a reasonable preference when you don't yet know what caused the error.

[Snapshot debugging - Business Central](#)

Alex: It is a reasonable preference, but Full needs translating. It records broader method context and source for the eligible code it encounters. It doesn't mean every variable value at every instant. Variable values are still captured at snappoints and exceptions. Snappoint is the slightly awkward name for a point where the recording collects state without stopping execution. Full capture can also materially increase resource use and the size of the recording, so it's intended for short, targeted scenarios rather than an automatic everything setting.

[Snapshot debugging - Business Central](#)

Emma: So my wish to spare the user another interruption doesn't automatically justify a much bigger capture. I'd rather have the developer explain what extra question Full is meant to answer. And if a value wasn't captured at the useful point, an AI explanation doesn't turn that missing value into observed evidence.

[Snapshot debugging - Business Central](#)

Alex: Exactly. The useful question is what the recording actually supports. The AI-assisted capture can make evidence collection easier, but diagnosis still involves separating what was observed from a possible explanation. A proposed fix is another step again. For our posting example, a captured failure might narrow the relevant code path. That helps the investigation; it doesn't by itself prove that changing one line will safely solve the business problem. You still need to verify the proposed correction.

[Snapshot debugging - Business Central](#)

Emma: Who is allowed to prepare this capture? The phrase describe it to an agent can make the whole thing sound as casual as asking a question in a chat window. But this is reaching into a Business Central session and collecting material from it.

[Snapshot debugging - Business Central](#)

Alex: The environment's MCP capability has to be enabled, and the caller needs the documented snapshot-debugging and attach-debug permissions. That includes the attach permission even when using this route for their own session. The actions use the signed-in person's identity and permissions; the agent isn't a separate permission shortcut. For a system owner, the practical preparation is to agree which authorized support people can use this workflow, and how they coordinate a reproduction with the person doing the work.

[Snapshot debugging - Business Central](#)

Emma: And the result needs a home that matches what it contains. A screenshot of an error and a recording with customer data aren't interchangeable attachments. I wouldn't want an easier capture button to turn into a habit of dropping every snapshot into whichever chat happens to be open.

[Snapshot debugging - Business Central](#)

Alex: Microsoft explicitly warns that snapshots can contain private customer data, and says to handle them under the relevant privacy and compliance policies and delete them when they're no longer needed. I'd make that part of the support handover itself: where the file goes, who can inspect it, and when it can be removed. That preserves the benefit of collecting better evidence without treating the evidence as harmless just because it came out of a developer tool.

[Snapshot debugging - Business Central](#)

Emma: For someone who's used snapshot debugging before, then, the attraction isn't a different recording format or a promise of instant repair. It's less manual coordination around starting, watching and collecting the capture. For the user, the improvement could be a more focused request: reproduce this particular problem now, while we're ready to learn from it.

[Snapshot debugging - Business Central](#)

Alex: Yes. The agent can prepare the capture, check its status, stop it and retrieve the recording through the supported tools, instead of the developer manually coordinating those commands. The recording still comes from the same snapshot engine. That continuity matters: you're improving the route to evidence you can inspect, not replacing the evidence with a fluent account of what might have happened. I'd judge the feature by whether it gets the right failure into an inspectable recording with less friction for the people involved.

[Snapshot debugging - Business Central](#)

Emma: That's where we'll leave it. What I like about today's changes is how close they sit to the work itself: helping a colleague pick up a record, making a product choice clearer, or giving support better evidence to investigate. And with sales agents, the useful question isn't how many you can have, but which kind of work each one should help move forward. The episode notes have the direct sources, release dates and technical detail for the part you want to explore. Thanks for listening to Central Perspectives from ITrators. Until next time.

[Add pictures to item variants to differentiate product options](#)

[Set up Sales Order Agent](#)

[Feature details for Business Central 2026 release wave 2](#)

[Snapshot debugging - Business Central](#)