

# Hub and Spoke Integration with Microsoft Dynamics AX

**AN INTEGRATION ROADMAP FOR MICROSOFT DYNAMICS AX™**

**White Paper**

March 2005

## USING THIS DOCUMENT

This document provides an overview of the integration capabilities for Microsoft Dynamics AX™ as they are specifically applied to the problems of hub and spoke integration. It is intended to:

1. Provide an introduction to common hub and spoke integration scenarios.
2. Describe the challenges in implementing these scenarios.
3. Provide an introduction to the tools and infrastructure available for hub and spoke integration out-of-the-box" with Microsoft Dynamics AX 3.0.
4. Provide an overview of additional integration solutions available from Microsoft that work well in conjunction with Microsoft Dynamics AX.
5. Describe the roadmap for enhanced hub and spoke functionality planned for Microsoft Dynamics AX 4.0 and beyond.

## DISCLAIMER

The uses of Microsoft Dynamics AX described in this document are intended as examples only and are not intended to describe the only way to use the functionality of Microsoft Dynamics AX 3.0 to accomplish hub and spoke integration scenarios.

Functionality described in this document that has not yet been released is subject to change without notice, and may be delayed or removed from future releases.

**This material is for informational purposes only.**

See the full legal disclaimer on the last page of this document.

# Contents

Executive Summary .....	4
Overview .....	5
Background .....	6
Data Sharing .....	6
Common Hub and Spoke Scenarios .....	9
Centralized Financials .....	10
Scenario Details and Challenges .....	10
Enabling Centralized Financials in Microsoft Dynamics AX 3.0 .....	11
Centralized Financials In Microsoft Dynamics AX 4.0 and Beyond .....	12
Centralized Procurement .....	13
Scenario Details and Challenges .....	13
Enabling Centralized Procurement on Microsoft Dynamics AX 3.0 .....	14
Centralized Procurement on Microsoft Dynamics AX 4.0 and Beyond .....	16
Distributed Fulfillment .....	17
Scenario Details and Challenges .....	17
Enabling Distributed Fulfillment on Microsoft Dynamics AX 3.0 .....	18
Distributed Fulfillment on Microsoft Dynamics AX 4.0 and Beyond .....	18

# Executive Summary

Hub and spoke implementations can have a number of benefits. Some of them are:

- **Simplicity** – Individual spokes are, by definition, only responsible to fulfill a portion of the company's overall data processing requirements. This means the applications can be smaller and simpler than monolithic applications, which must account for every requirement in every location.
- **Best of Breed Solutions** – Spokes can run the software that best meets their specific local requirements. Even when all spokes run the same software, it can often be configured or customized to best meet local requirements.
- **Local Control and Accountability** – Some companies find it easier to do most of the administration and support locally, because it minimizes language and time zone barriers. The increased local control also improves local accountability.
- **Better Performance** – Local applications running on local servers with smaller user counts can yield huge performance benefits.
- **Lower Cost** – Implementation costs are reduced because the spokes run simpler software, which is quicker and easier to implement. Upgrades are easier and less painful when spokes can be upgraded one at a time.
- **Keep Existing Applications** – As companies grow, hub and spoke can be an effective way to organize and expand their computer resources without replacing all their existing software. Collecting and consolidating information at the hub provides management with the visibility they need to the overall operation, but without the pain and distraction of replacing their core applications.

## Overview

For the purposes of this paper, hub and spoke is defined as business processes that are shared between a central managing "hub" and departmental or business unit "spokes", each running its own software implementation, and all within a single company or corporation. Many types of business-to-business (B2B) interaction are referred to as hub and spoke, especially in supply chain circles. And to be sure, many of the concepts apply readily to B2B scenarios; however, this paper will not address the contractual obligations or stringent security requirements that usually accompany B2B implementations.

Because it would be impossible to provide a comprehensive survey of hub and spoke scenarios in this type of document, this paper will focus on three common sample scenarios, namely:

- Centralized Financials
- Centralized Procurement
- Distributed Fulfillment

They were chosen because the concepts, challenges, and suggested solutions should readily translate to other hub and spoke scenarios.

Each scenario will be described first in general terms. This includes a basic description of the business process, master data that is shared between the hub and spokes, transactions that must be passed between them, and software features and functionality that are required at the hub and at the spokes.

Next, the paper will discuss how Microsoft Dynamics AX 3.0 can be used as the hub and as spokes in each scenario. Existing Microsoft Dynamics AX functionality that is useful for the scenario is highlighted. In areas where Microsoft Dynamics AX does not provide a comprehensive out-of-the-box solution, customizations and software extensions are suggested.

Finally, possible future enhancements to Microsoft Dynamics AX to make it even better suited to support these scenarios are described. The analysis is divided into features for moderately complex and then highly complex scenarios. These should not be interpreted as commitments to provide any specific functionality in the future. Rather, they should be viewed as future opportunities for Microsoft and its partners and customers.

## Background

Microsoft Dynamics AX is a multiple language, multiple currency enterprise resource planning (ERP) solution with core strengths in manufacturing and e-commerce, and strong functionality for the wholesale and services industries. With its comprehensive functionality, it is an excellent all-in-one solution to help mid-size and large companies across the globe seize opportunity and gain competitive advantage.

In cases where all-in-one solutions are not practical, Microsoft Dynamics AX is also an excellent team solution. For example, it can be integrated as departmental “spokes” to extend an existing enterprise “hub” system. These scenarios are common when companies need to bring new offices or departments on-line quickly, but are handcuffed by massive investments in their existing ERP infrastructure. These existing systems are often difficult, time-consuming, and expensive to customize or extend, or might simply be out of capacity.

Microsoft Dynamics AX also serves well as the hub in hub and spoke deployments. In geographically dispersed companies, for example, these deployments are often motivated by challenges with network bandwidth or reliability, time zones, language, or other considerations.

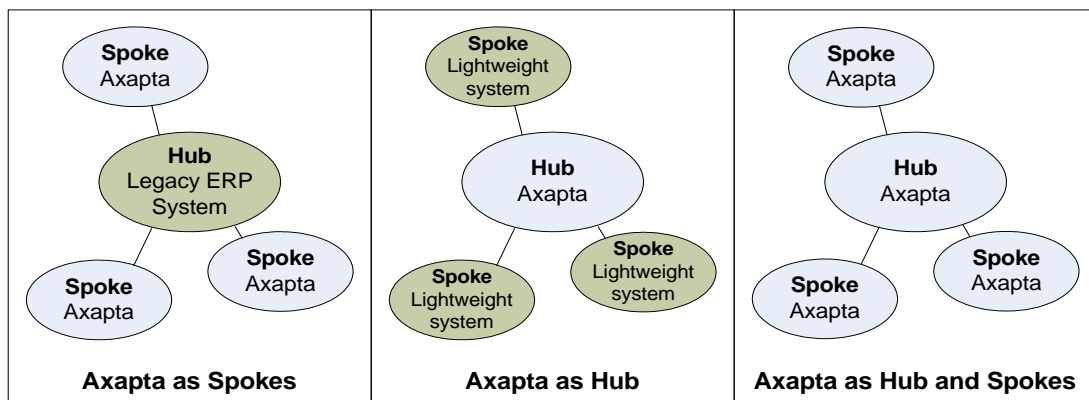


Figure 1 - Microsoft Dynamics AX as Hubs and Spokes

### DATA SHARING

Within any hub and spoke implementation, there are three classes of data that must be shared between the hub and its spokes:

- Working documents
- Transactions
- Reference data

### Working Documents

Working documents are what drive business processes. Examples are sales orders or purchase orders. They typically evolve over time as the business process moves through completion. It's also valuable to capture snapshots of working documents at various points in the business process. For example, sales order or purchase order confirmations capture snapshots of the orders after they are finalized. Working documents:

- Relate to specific instances of a business process (for example, sales orders or purchase orders) or subflow (for example, pick tickets or packing slips).
- Are created and then change state over time as dictated by the business process workflow.
- Are time-sensitive. Working documents represent the current status of a particular instance of a business process workflow.

Working documents often are:

- Electronic representations of work items that businesses have historically maintained on preprinted, numbered forms, such as invoices or purchase orders.
- Represent legally binding agreements between the organization and external trading partners.
- Monetarily valuable, even when they are not legally binding.
- Time critical, with expiration dates.
- Historically valuable. That is to say, there is value in retaining access to closed Documents.

In the hub and spoke scenarios described in this paper, a single node has clear “ownership” of the working documents. Documents are created on the hub or in the spokes, and continue to be managed by the same node throughout the document lifecycle. Other nodes might receive status information from the document owner but will not directly make changes to the document.

### Transactions

Where working documents drive business processes, transactions capture the execution details. For example, when an invoice is posted as part of the sales order fulfillment process, transactions record the change to the customer’s accounts receivable and capture the necessary updates in the general ledger. Transactions:

- Convey information about an event.
- Are the results of posting, usually to a ledger.
- Are not time-sensitive in the same way as working documents. They are posted at a particular point in time and are absolutely true from that point forward; even after they are archived or deleted. They do not expire.
- Never represent (by themselves) legally binding agreements.

Transactions:

- Can be multidimensional. For example, an inventory adjustment transaction could have additional dimensions that define the warehouse bin locations, lots, or serial numbers that were adjusted.
- Can have monetary value.
- Can consist of multiple transactions which may be (but are not required to be) aggregated into batches. For example, multiple shipment updates can be aggregated into a single inventory update batch.
- If posting results in transactions, separate transactions can be created to correspond to each detail line in the source data. This is referred to as “posting in detail”. Alternatively, multiple detail lines can be aggregated into a single transaction, which is referred to as “posting in summary”. If posting results in multiple transaction types (e.g., posting an inventory adjustment results in inventory transactions and general ledger transactions), some transaction types might be posted in summary while others are posted in detail.
- Every document or master file update does not necessarily result in a transaction.

In hub and spoke scenarios, transactions are used by the owner of working documents to communicate events to other nodes. These events often trigger new business processes in the receiving nodes. For example, in the distributed fulfillment scenario described later, shipping decisions made by the hub (owner of a sales order working document) are communicated to the nodes as pick list transactions.

### Reference Data

Reference data, or “master data”, provides the context for business processes. Examples of reference data are customer, vendor, or item master tables. Like working documents, reference data changes over time. Reference data:

- Is not specific to a business process instance, but is shared across many instances.
- Conveys context information that is necessary to carry out business processes.
- Changes over time. Therefore, it can be out of date.

Reference data:

- Often describes physical aspects of the world the business operates in. For example, customer master records describe physical characteristics of the customer such as name, address, or contact information.
- Usually does not have “state”. (The most common exception is Active/Inactive flags.)

In hub and spoke scenarios, reference data has its own special set of problems. These arise because reference data often does not have clear ownership boundaries. For example, in the distributed fulfillment scenario discussed later, both the hub and the spokes have legitimate claim to the inventory master table. Solutions that arbitrarily designate one node to own the table and manage its maintenance will be awkward and cumbersome in real world implementations.

# Common Hub and Spoke Scenarios

The range of documents, transactions, and reference data that might be required in hub and spoke implementations is immense. Rather than trying to address this broad range in generalities, this paper will focus on three specific common hub and spoke scenarios. They are:

## **Centralized Financials**

In this scenario, financial transactions such as general ledger journal entries are created by the spoke systems and transmitted to the hub.

## **Centralized Procurement**

In a typical implementation of centralized purchasing, requisitions (or purchase requests) are submitted by the spokes to a centralized purchasing department at the hub. The hub places the purchase orders with the vendor and communicates status back to the spokes. Receiving can be performed either at the hub or the spokes. Invoice matching is performed at the hub.

## **Distributed Fulfillment**

In this scenario, customer orders are captured by a centralized order processing department, however, they are fulfilled by shipping from one or more spoke warehouses. Shipping information is communicated back to the hub for invoicing. Availability information is also regularly communicated from the spokes to the hub.

# Centralized Financials

In these scenarios, the spokes are organized as autonomous business units, with their day-to-day activities being relatively self-contained. For example, they might have their own customers and process their own orders. As financial results are generated by each business unit, they are sent to the hub, where they are combined and consolidated. Factors that motivate companies to select this topology include:

- Acquisitions – This is a relatively easy way to merge acquired companies who already have their own processes and IT systems in place.
- Business Requirements – Sometimes the products and services provided by different business units are so unique that it is impractical to support all their IT requirements with a single ERP implementation.
- Geography – Language, localization, and time zone issues make it difficult to support a single centralized system.
- Performance and Scalability – In larger organizations, this can be an effective way to manage performance for many users across a slow and unreliable WAN.

## SCENARIO DETAILS AND CHALLENGES

As the spokes carry out their business processes, the local ERP system creates financial postings in the form of general ledger journal entries. The journal entries are transmitted to the hub, where they are combined with those from other spokes and consolidated into overall financial statements.

### Charts of Accounts

For centralized financials to make sense, the hub and all the spokes should use the same account structure in their charts of accounts. It's easiest to do this by maintaining a master chart of accounts at the hub and treating it as reference data for the spokes. This is practical because the chart of accounts changes infrequently.

If it is not possible to maintain the same chart of accounts at the hub and all the spokes, then the accounts must be mapped as they move from the spokes to the hub. If the accounts have a one-to-one relationship between the spoke and hub, the mapping does not add much complexity. However, it's possible that the chart of accounts is structurally different between the hub and spoke. For example, the hub might have their sales account structure broken down by geography (such as sales regions) but some of the spokes might break them down by customer type. Because these breakdowns are often hard coded in legacy applications, this situation is relatively common when the hub and spokes are running different software. In these cases, the logic required to map between the spokes and hub can get quite complex and is seldom completely satisfactory.

Also, when new accounts are created at the hub and transferred, there might be other configuration work that needs to be synchronized at the spokes. For example, if a new account is created to capture costs for a new class of inventory items, the inventory class table on the spoke would need to be updated to reflect the new class and the new inventory items will need to be created. If these synchronization problems are too frequent to manage manually, it is necessary to enable spokes to create their own accounts locally and send updates to the master chart of accounts on the hub.

### Budgets

Although technically not a part of the scenario, companies who implement centralized financials often want to incorporate their budgets as part of the information that is shared between the hub and spokes. In its simplest form, they can develop the budgets offline and key them into the database on the hub. As the chart of accounts reference data is sent to the spokes, the budgets are included. Alternatively, budgets can be developed on the spokes and collected and consolidated at the hub.

### **Detailed versus Summary Postings**

The journal entries created at the spokes are fine-grained, providing full details about each transaction. This is helpful for drill-down and audit purposes. At the hubs, however, these details have less value because the backing transactions such as the original invoices or AP checks are not available at the hubs. For that reason, it might be better to summarize transactions as they are moved from the spokes to the hub. This can be straightforward if the software on the spokes supports sub ledgers and summary reporting. If it doesn't, then the summarization must be accomplished in the middleware that does the data transport.

### **Currency**

If the spokes use a different currency to the hub, then currency conversions are necessary as journal entries are passed. This introduces the normal challenges of managing exchange rates and dealing with the rounding issues as values are converted from the spoke currency to the hub currency.

### **Triggering Updates**

As accounts are added, modified or retired in the chart of accounts on the hub, the changes must be pushed to the spokes. Depending upon the frequency and timing of updates, it can be satisfactory to refresh the entire chart of accounts on the spokes on a regular (for example, daily, weekly, or monthly) basis as a batch. If changes need to be propagated in real time, then the hub software must capture those changes as events and send "update" transactions to the spokes.

Similarly, the timing dictates the strategy used to send journal entries from the spokes to the hub. A common approach is for the spokes to close their books on a particular day of the month and send all the journal entries for the month as a batch. If more frequent updates are required, they can be triggered as journal entries are posted to the local general ledger.

### **Capturing Transactions on the Hub**

As journal entry transactions are received from the spokes, they must be captured and posted to the general ledger on the hub. Some companies want to manually review the batches before releasing them to be posted to the general ledger. In more complex scenarios, the hub and spokes might want to collaborate on "trial" financial statements before finalizing.

## **ENABLING CENTRALIZED FINANCIALS IN MICROSOFT DYNAMICS AX 3.0**

Although Microsoft Dynamics AX 3.0 does not provide a turnkey solution for hub and spoke centralized financials, there are many aspects of its architecture and feature set that make it well suited for these scenarios.

### **Microsoft Dynamics AX as the Hub**

One aspect of the Microsoft Dynamics AX architecture that makes it ideal for hub and spoke centralized financial scenarios is the strong journal-based paradigm. When Microsoft Dynamics AX is the hub, the general journal provides a logical place to capture the journal entries transmitted by the spokes. The out-of-the-box reporting and approval logic makes it easy to manually review the journal entries before posting them.

In addition, the Microsoft Dynamics AX General Ledger has excellent support for multiple currencies. This means that the journal entries can be captured from the spokes in their native currency. The Microsoft Dynamics AX infrastructure can handle the details of exchange rates and currency conversion.

Microsoft Dynamics AX has a simple account structure, rich financial dimensions, and flexible reporting that enable it to operate with fewer accounts than many of the systems it will connect to. For example, in many systems departments are hard coded into the General Ledger account structure. With Microsoft Dynamics AX, departments are a financial dimension and do not need to be coded into the base account structure. Because the dimensions are not hard coded in the Microsoft Dynamics AX account structure, it reduces the structural mismatches between Microsoft Dynamics AX accounts and the account structures on the spokes. This simplifies the process of mapping spoke accounts into the hub chart of accounts.

### **Microsoft Dynamics AX as a Spoke**

Many of the benefits listed above also help to make Microsoft Dynamics AX an excellent spoke for hub and spoke centralized financial scenarios.

The journal-based paradigm provides a place from which to trigger the transmission of journal entries to the hub. That is, as journals are posted on the local system, the postings can be captured as events that cause the journals to be transmitted.

Alternatively, the financial reporting in Microsoft Dynamics AX can become the basis for pushing financial information to the hub, either summarized or in detail. Moreover, the batch scheduling capabilities could be used to execute the reports at night, during times of low system usage.

If the hub does not have good support for multiple currencies, the multicurrency support in Microsoft Dynamics AX could be used to push out the financials in the hub's base currency, even though the transactions are executed in the spoke's local currency.

### **CENTRALIZED FINANCIALS IN MICROSOFT DYNAMICS AX 4.0 AND BEYOND**

In the future, Microsoft Dynamics AX will be enhanced to more easily support hub and spoke scenarios and to support moderately to highly complex implementations. Some of the improvements that can be provided in future versions of Microsoft Dynamics AX are:

#### **Moderately Complex Solutions**

- Output journals and financial reports in XML format
- Create journals from XML
- Provide a rich system-to-system framework to facilitate transporting data between the hub and spokes
  - Reliable delivery
  - Secure environment
  - Better integration to middleware such as BizTalk® Server
- Provide a powerful web service interface to make it easier to develop custom solutions
- Automatically trigger the transmission of journals from user actions such as posting

#### **Highly Complex Solutions**

The items listed above for moderately complex solutions involve enhancing Microsoft Dynamics AX to the Microsoft Dynamics AX infrastructure. To support even more complex solutions, changes to the Microsoft Dynamics AX architecture and business logic will be required. Some of the solutions that can be envisioned are:

- Distributed budgeting and approvals
- Collaborative closing process
- Replicating consolidated financials back to the spokes

# Centralized Procurement

In these scenarios, a centralized purchasing department is responsible for collecting purchasing requirements from several offices, departments, or warehouses and the combining them to make the most efficient purchases. In the case of materials, supplies, and capital purchases, the purchasing requirements usually take the form of requisitions. In the case of trade goods, buyers might have the responsibility for determining purchase requirements based on current quantities available and usage history.

## SCENARIO DETAILS AND CHALLENGES

### Materials, Supplies, and Capital Purchases<sup>1</sup>

For these purchases, a requisition is created at the spokes. Before being sent to the hub, it might need to go through one or more local approval steps. After being approved, it is transmitted to the hub, where additional approvals might be required. After it is approved, it is combined with other requisitions and either sent out for bid or ordered from an approved vendor.

Reference data required at the spokes includes lists of approved vendors, and item catalogs. The approval rules can also be managed at the hub, so if local approvals are required then the rules must be propagated out to the spokes.

The working documents in this scenario are interesting. In an ideal world, the requisition would be created at the spokes, and when it is approved and sent to the hub, the hub and spoke would share ownership. Users on the hub would be free to edit the requisition and update status as it moves through the purchasing process. The spokes would be free to make changes in response to new requirements or new information from the buyer or vendor. As a practical matter, however, dual ownership is difficult to implement and administer.

One approach to simplify this arrangement is to move the ownership at the time the requisition is released to the hub. As it is being created and locally approved, the spoke owns the requisition. If the hub has access to it at all during this stage, it is treated as view-only reference data. After the requisition is released, ownership is transferred from the spoke to the hub. From that point on, the hub is free to make changes and updates. These are propagated back to the spoke as reference data. During this phase, the spoke is not allowed to make changes. If changes are required, these can be handled off-line with telephone calls, fax, or e-mail.

### Trade Goods<sup>1</sup>

If the hub is responsible for making purchasing decisions about raw materials or finished trade goods, a completely different set of reference data is required. The hub usually owns the inventory master data, which is sent to the spokes periodically as reference data. The spokes own the information regarding quantities available. They might also own the statistics about inventory usage which are required to make good buying decisions. Either the hub or the spokes can own the production plans.

As purchase orders are created, they can be sent to the spokes as reference data.

### Multi-tiered Production or Distribution

In multi-tiered production or distribution environments, some of the nodes can be both hubs and spokes. For example, the diagram below depicts a 3-tiered distribution environment with a central master distribution center, regional distribution centers and local warehouses.

In these environments, the middle tier acts as a hub when transacting with the local warehouses, and as a spoke when transacting with the master distribution center. The spokes might own their own quantity on hand or usage history information, which is propagated to the middle tiers as read-only reference data. That same data, when it is aggregated for all the warehouses in the region, is owned by the regional distribution center. It is propagated to the master distribution center as reference data.

---

<sup>1</sup> Even though these examples focus on the procurement of tangible goods, the concepts are the same for services.

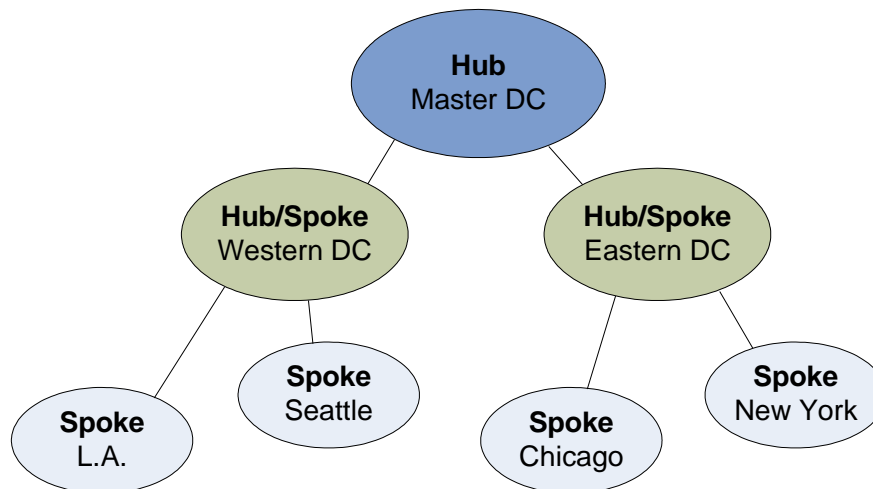


Figure 2 - Multi-Tiered Distribution Example

### Receipts

After the order is placed, the goods can be sent by the vendor directly to the spoke location, or they can be sent to the hub (a distribution center). If they are sent directly to the spoke location, the spoke acknowledges the receipt to the hub so that the hub can process the payment. Even in highly automated environments, this is often accomplished by sending copies of the vendor's packing slip to the hub for reconciliation.

If the goods are shipped to the distribution center, then the distribution center will combine the goods with other orders and forward them to the spoke location. In well-automated systems, status information in the form of advanced shipping notices is transmitted to the spokes so they can be prepared to receive the shipments when they arrive.

### Status Information

Throughout this process, from the time requisitions are transmitted to the hub until the goods are received and acknowledged at the spoke and the vendor's invoice is processed for payment, spokes need the ability to inquire on the status of their orders.

In the simplest implementations, status is handled out-of-band. That is, electronic status updates are not sent to the spokes at all. Instead, the spokes must request the status of particular requisitions or orders by phone, fax or e-mail. When combined with daily or weekly status reports of all open orders, this can be quite satisfactory in lower volume environments.

In moderately complex solutions, status information can be made available to the spokes through browser-based inquiry tools. The spoke must still request the status for selected requisitions or orders, but the inquiry tools make this relatively quick and efficient. The browser-based tools usually enable the spokes to see reports of all their open orders and also drill into the details of particular orders. When combined with good processes and procedures at the spokes, this can work well even for much higher volume environments.

In the fullest implementations, status information is automatically propagated to the spokes each time there is a change in status at the hub. This has the benefit that event-driven alerts can be used at the spokes to automatically notify local users of critical status changes. In the highest volume environments, where the ability to manage by exception is critical, this level of automation is extremely valuable.

### ENABLING CENTRALIZED PROCUREMENT ON MICROSOFT DYNAMICS AX 3.0

The Microsoft Dynamics AX architecture is well suited to participate in hub and spoke scenarios for procurement. For example, its multilocation inventory, and journal-based updates provide a convenient structure for capturing and communicating inventory quantities on hand and available.

### Microsoft Dynamics AX as the Hub

For simple implementations of purchases for materials, supplies and capital items, the Microsoft Dynamics AX purchase order module can work well. Requisitions received from the spokes are used to create purchase orders. As the purchase orders progress through the procurement process, their status can be kept up to date either manually, or by custom-developed communication channels with the vendor, such as EDI. As already discussed, this status information can be communicated back to the spokes through phone, fax, e-mail, browser based status inquiries, or custom-developed communication channels. In the latter case, the event-based programming model in Microsoft Dynamics AX makes it relatively easy to detect status changes as they are made to the purchase orders.

Solutions become more complex if procurement of trade goods is centralized at the Microsoft Dynamics AX hub. This is because the tools in Microsoft Dynamics AX that are available to automatically create purchase orders assume that the inventory availability and usage history reside on the local computer. For them to work properly, all of this information must be captured in the local inventory tables as reference data. Some of the information that must be captured is:

- Inventory Quantities on Hand
- Inventory Sales History
- Open Sales Orders
- Open Production Orders and Production Schedules

If the hub is also doing centralized order taking (refer to the next scenario regarding distributed fulfillment), then much of this data resides on the hub system as working documents.

### Microsoft Dynamics AX as the Spoke

As a spoke, Microsoft Dynamics AX allows the creation of purchase orders for materials, supplies, and capital items; to serve as requisitions when sent to the hub. The two primary challenges center around vendors and items. For example, it's generally the responsibility of the centralized purchasing department at the hub to select the best vendors, even though a vendor ID is required on the Microsoft Dynamics AX purchase orders. As POs are transmitted to the hub, the vendor ID must either be stripped off or ignored by the hub.

For common supplies such as paper or toner cartridges, hubs often maintain standard lists or order forms. If the number of items is large, it is probably convenient to store these items as reference data in the Microsoft Dynamics AX inventory. They should be configured so that they can be easily filtered out of inventory reports for trade items. A good approach is to create special model and dimension groups for these items.

For special order items, or when there is no list of standard items, creating a single "special order" item can reduce the amount of coordination necessary between the hub and spokes. The item details and specifications are manually entered in the Text and External fields on the line items or attached to the lines using the Microsoft Dynamics AX document management features.

The problem is larger when the hub manages the procurement of trade goods. Inventory master data or catalogs are often owned by the hub, so items must be created as reference data in the inventory master tables. In this case, bills of material can also be owned by the hub and transferred to the Microsoft Dynamics AX spokes as reference data. Also, depending upon the strategy for order capture and fulfillment, the sales orders, production orders, and demand or utilization history can be owned locally or by the hub.

As purchase orders are placed at the hub, they are replicated to the Microsoft Dynamics AX spokes as reference data. Ideally, changes in purchase order status are updated to the reference purchase orders in near real time, although simpler options have already been discussed.

The journal structure of Microsoft Dynamics AX makes it relatively easy to capture receipt information and send it to the hubs. For example, inventory receipts can be entered against the reference purchase orders. When packing slips are posted, they can be sent to the hub to serve as receipt notifications.

There are, however, two complicating factors. First, to post inventory receipts, it's necessary to make changes to the purchase order lines such as updating the Receive Now field. Also, Microsoft Dynamics AX will want to

update the purchase order status and quantities on hand as the receipts are posted. Therefore, a shared ownership strategy for purchase orders is required. The hub owns the creation of purchase orders and some details of the lines, such as item numbers, quantities on order, or expected dates of arrival. The spoke owns other details such as current status and quantities received.

Policies to lock the purchase order down before the first receipt can help to simplify this shared ownership. For example, allowing part numbers to change or lines to be added or deleted after the purchase order has been partially received can get quite complex. By restricting them, simpler synchronization strategies can be employed while minimizing hardship to the purchasing processes.

## **CENTRALIZED PROCUREMENT ON MICROSOFT DYNAMICS AX 4.0 AND BEYOND**

To be a premier hub and spoke product, especially in the manufacturing and distribution industries, Microsoft Dynamics AX can be enhanced to provide better support for centralized procurement scenarios. In the future, Microsoft should provide moderately and highly complex solutions.

### **Moderately Complex Solutions**

The future requirements already described in relation to centralized financials apply equally to the problems in this section. However procurement documents tend to be more complex than those required for centralized financials. For that reason, additional requirements for moderately complex centralized procurement solutions include:

- Out-of-the-box integration capabilities for complex documents such as purchase orders.
  - Represent entities comprised of many Microsoft Dynamics AX tables as consolidated “logical” documents
  - Reduce the need for integration developers to understand underlying Microsoft Dynamics AX table structures
  - Automatically invoke Microsoft Dynamics AX business logic to validate documents as they are submitted to the Microsoft Dynamics AX database
  - Automatically invoke custom business logic as documents are submitted to the Microsoft Dynamics AX database
- User-configurable alerts to keep hubs and spokes aware of changes in PO status
- A rich framework for browser based applications that can be used to publish order status information to outlying offices.

### **Highly Complex Solutions**

Some of the changes that can be envisioned for the Microsoft Dynamics AX architecture and business logic are:

- Requisitions with Bid Management
- Distributed Budgeting and Approvals
- Distributed Available to Promise (ATP) and Capable to Promise (CTP)
- Centralized Inventory and Replenishment Planning

# Distributed Fulfillment

In distributed fulfillment scenarios, incoming orders are captured and managed centrally, while fulfillment responsibilities are pushed out to the spoke locations. Two common reasons to select this topology are:

- Geography – Move fulfillment closer to the customers or suppliers
- Specialization – Consolidate expertise in particular product lines or processes

## SCENARIO DETAILS AND CHALLENGES

After sales orders are captured at the hub, fulfillment information is pushed to the spokes. In some implementations, the hub is relatively uninvolved with the fulfillment process. Orders are subdivided by fulfillment location and forwarded to the hubs either immediately after they are captured, or some predetermined time before they have been requested to ship. When the orders are sent, the spokes make the final decisions about the prioritization of manufacture and shipment, etc. In other implementations, the hub directs the fulfillment process by prioritizing orders, and scheduling production and shipment. For the purposes of this paper, the former implementations are referred to as “distributed planning” and the latter as “centralized planning” strategies.

### Distributed Planning

With distributed planning strategies, orders are handled on the hub similarly to how drop shipments are handled. The hub captures the orders and tracks status. It places drop-ship purchase orders with the spoke and waits for notification that orders have shipped before invoicing the customer.

At the spokes, the drop-ship purchase orders are used to create local sales orders; which are processed and fulfilled in exactly the same manner as single-location companies. The PO is received from the hub. The spoke is responsible to procure the raw materials, and manufacture and ship the item. At the end of the process, the spoke “invoices” the hub for the goods shipped, which also serves as a shipment notification to the hub.

For these implementations, a common understanding about inventory part numbers and customer shipping information is required. The usual approach would be for the hub to own the item master tables, which are pushed to the spokes as reference data. The spokes are responsible for their own inventory. This information can be updated periodically back to the hub to aid with order taking and customer service. Alternatively, it can be made available to the hub through conventional reports or real-time browser-based inquiry.

Interestingly, in this arrangement both hub and spokes maintain sales orders as working documents. After orders are captured at the hub and corresponding purchase orders are created, the PO confirmation transactions are forwarded to the spokes. In cases where a single customer order will be fulfilled by multiple spoke locations, each spoke only receives a PO for the items it is expected to ship. The hub, then, treats each spoke as a vendor.

The spokes treat the hub as a customer. The PO confirmation transactions from the hub cause sales orders to be created locally. Unlike reference data, these sales orders are not duplicates of the original order. Their “sold-to” information points back to the hub and the final customer is merely a shipping address. Also, pricing on these sales orders reflects the amount that will be charged back to the hub – not the final selling price to the end customer.

After the order is fulfilled, an invoice is created and sent to the hub. This invoice serves as a shipping notice to trigger invoicing the customer. From an accounting perspective, the spoke’s accounts receivable amount is exactly offset by the hub’s accounts payable amount.

In an alternative form of this scenario, the goods are not drop-shipped directly from the spokes to the customer, but rather are shipped to the hub. The hub consolidates these items with those from other spokes into a single shipment. This might be done, for example, to consolidate the customer’s entire order into a single shipping container for overseas shipment.

Distributed planning strategies can be attractive because most of the required software functionality at both the hub and spokes already exists in most ERP systems.

### **Centralized Planning**

With centralized planning strategies, there is only one order. The hub is responsible for tracking the order, inventory availability, and plant capacity similarly to normal multiwarehouse implementations. The spokes are treated as warehouses for purposes of procuring raw materials, creating production plans, and scheduling shipments. As production and shipping orders are created on the hub, they are sent to the spokes to carry out the work. The spokes notify the hub as each step is completed.

These strategies are especially attractive when planning and coordination across multiple spoke locations is important. Although the software required at the hub is sophisticated, the software at the spokes can be very simple. In fact, in the simplest implementations the spokes don't run any software at all. The transactions are managed by faxing production orders back and forth between hub and spokes.

### **Multi-tiered Production and Fulfillment**

When production and fulfillment are organized into complex networks similar to those already discussed for multi-tiered procurement, the implementation strategies are much more difficult. Distributed planning, with production spread across autonomous nodes, can be the simplest to implement from a software standpoint, however changes and exceptions must ripple through the network in a relatively unmanaged way. This leads to poor reaction times, with reduced flexibility and poor customer service.

Centralized planning and management strategies require even more sophisticated production and distribution planning software at the hub, however the software required at the spokes remains quite simple. For that reason, this is often the best choice when the production and fulfillment network becomes complex.

## **ENABLING DISTRIBUTED FULFILMENT ON MICROSOFT DYNAMICS AX 3.0**

Because the distributed planning model mirrors single-site order management so closely, this is probably the easiest to implement with Microsoft Dynamics AX 3.0. As the hub, when orders are captured from the customer, Microsoft Dynamics AX already provides the means to create corresponding drop-ship purchase orders. As a spoke, Microsoft Dynamics AX processes the orders the same way as "normal" drop-ship orders. Settlement between the AP voucher on the hub and the AR invoice on the spokes can be processed manually or custom reconciliation processes can be developed.

In the centralized planning model, the Microsoft Dynamics AX multi-warehouse capabilities can be used when it serves as the hub. Each spoke is configured as a warehouse, with Microsoft Dynamics AX managing inventory levels and commitments. When production orders or shipping orders are created, they become the documents that will be communicated to the spokes. Microsoft Dynamics AX is not so well-suited to operate as a spoke in the centralized planning model. That is because it is specifically designed to optimize production and fulfillment for open orders; however the centralized planning model calls for it to execute 'as instructed'.

## **DISTRIBUTED FULFILMENT ON MICROSOFT DYNAMICS AX 4.0 AND BEYOND**

As Microsoft Dynamics AX becomes more sophisticated in the areas of distributed fulfillment, some of the features that can be provided by Microsoft are:

### **Moderately Complex Solutions**

All the features already mentioned in the previous two scenarios certainly apply to moderately complex distributed fulfillment scenarios. In addition, it would be easier to implement these scenarios if Microsoft Dynamics AX were to provide the following specialized features in future releases:

- Out-of-the-box support for electronic purchase order transmission and receipt
- Out-of-the-box support for electronic invoice receipt and settlement
- Enhancements to the inter-company features to support departments that span physical Microsoft Dynamics AX implementations

- Enhancements to Enterprise Portal to make it easier to monitor inventory and order status across multiple spoke implementations

### Highly Complex Solutions

The most demanding scenario for Microsoft Dynamics AX is acting as the hub in multi-tiered, centralized planning scenarios. To support these, Microsoft Dynamics AX must be enhanced to support complex order nets with:

- Distributed available-to-promise (ATP) and capable-to-promise (CTP) calculations.
- The ability to optimize production based on sophisticated customer-defined planning rules and algorithms.
- The ability to automatically respond to most changes and exceptions, allowing production and fulfillment planners to manage by exception.

The information contained in this document represents the current view of Microsoft Corporation on the issues discussed as of the date of publication. Because Microsoft must respond to changing market conditions, it should not be interpreted to be a commitment on the part of Microsoft, and Microsoft cannot guarantee the accuracy of any information presented after the date of publication.

This white paper is for informational purposes only. MICROSOFT MAKES NO WARRANTIES, EXPRESS, IMPLIED, OR STATUTORY, AS TO THE INFORMATION IN THIS DOCUMENT.

Complying with all applicable copyright laws is the responsibility of the user. Without limiting the rights under copyright, no part of this document may be reproduced, stored in or introduced into a retrieval system, or transmitted in any form or by any means (electronic, mechanical, photocopying, recording, or otherwise), or for any purpose, without the express written permission of Microsoft Corporation.

Microsoft may have patents, patent applications, trademarks, copyrights, or other intellectual property rights covering subject matter in this document. Except as expressly provided in any written license agreement from Microsoft, the furnishing of this document does not give any license to these patents, trademarks, copyrights, or other intellectual property.

© 2005 Microsoft Corporation. All rights reserved.

Microsoft Dynamics AX, BizTalk are either registered trademarks or trademarks of Microsoft Corporation or Microsoft Business Solutions ApS in the United States and/or other countries. Microsoft Business Solutions ApS is a subsidiary of Microsoft Corporation.

The names of actual companies and products mentioned herein may be the trademarks of their respective owners.