MANAGE

Microsoft Dynamics NAV

Building Industry Business Success on Microsoft Dynamics NAV

Best Practices for Business Processes and Tools

White Paper

November 2008

www.microsoft.com/dynamics/nav

Microsoft Dynamics NAV

Table of Contents

Introduction

This white paper documents the best practices and tools available for developing vertical or industry solutions based on Microsoft Dynamics® NAV. The paper is for Microsoft Dynamics NAV partners who want to build vertical or industry solutions using their industry domain experience and for value-added reseller partners who want to learn more about end-customer customization tools.

After reading the paper, you will have a better understanding of:

- The vertical or industry product development process.

- Microsoft Dynamics NAV tools you can use in this process.

- The requirements for packaging a vertical or industry solution for a specific market segment.

This paper was written based on interview and feedback from a number of Microsoft Dynamics NAV partners. We would like to thank the following people for contributing with input:

- Matthías Matthíasson, Landsteinar Strengur

- Andreas Voss, Tectura

- Michael Kulik, Digital Vantage Point

- Richard Postborg, Cornator

- Johnny Madsen, Cornator

- André Overrem, 4PS

- Rainer Fischer, Impuls Informationsmanagement GmbH

- Gerhard Winter, Tegos GmbH

- Cornelis Bosch, To-Increase bv

If you have questions or require more information after reading this document, please contact Eva Sachse, Senior Product Manager, Microsoft Dynamics NAV Global Product Management at ebjerreg@microsoft.com.

Getting Started

To get started, download Microsoft Dynamics NAV from MSDN Subscriptions at

http://msdn.microsoft.com/en-us/subscriptions/default.aspx

Executive Summary

This white paper documents the tools and process information available to assist you in developing a vertical or industry solution built on Microsoft Dynamics NAV. You can use this paper as a guide to organizing your company's resources to build the solution more effectively.

The paper is based on the Microsoft® Solutions Framework (MSF), a proven methodology for delivering business-driven solutions, and follows the MSF Team Model, outlining the tools available for your team members.

For more information, read the MSF white papers posted at

www.microsoft.com/downloads/details.aspx?familyid=A71AC896-1D28-45A4-880C-8B0CC8265C63&displaylang=en.

If your time is limited, start by reading the "MSF Team Model" and the "MSF Process Model" papers.

Product Thinking

For a Microsoft Dynamics NAV partner, building a vertical or industry solution requires a change in approach from the typical customization activities to a focus on product development. This approach usually requires more discipline than typical one-off customizations. For example, you must consider the efforts needed to develop a solution that meets the needs of the particular vertical or industry industry and organize a team to manage the development, packaging, and maintenance of the solution.

The move to develop a vertical or industry solution may have resulted from an analysis of your existing customer base. If you decide to build a vertical or industry solution because you have been successful selling to that vertical or industry market by making several customizations, then you need to consider the individual needs of the customer versus the greater generic needs of the industry.

Be careful to avoid adding customizations in the solution without a proper analysis of the features needed for all the customers in the target industry, and do not take customizations directly from a customer site without reengineering them to fit into the core of your vertical or industry solution. Build additional customer customizations on top of your solution so that you minimize the number of unique customer-specific customizations.

Microsoft Dynamics NAV partners who have developed successful vertical or industry solutions have a separate team focused on product development and another team focused on implementation consulting. Having separate teams enables each to concentrate on solving the respective challenges of development and implementation.

Methodology

Delivering a software project on time and within budget requires careful management. This paper provides practical advice to help you manage the product development process.

Consider using a methodology that will be easy to implement in your organization. A number of methodologies or frameworks are available, and you may already work with one for implementation consulting. The Microsoft Solutions Framework is based on practices proven at Microsoft and provides people and process guidance to help teams and organizations become more successful in delivering business-driven technology solutions to customers. The MSF components discussed in this white paper can be applied to packaged application projects.

Microsoft Dynamics offers implementation best practices that can be deployed within the MSF Team Model. Microsoft Dynamics Sure Step is a proven methodology designed to help ensure fast, reliable implementation, migration, and upgrades for Microsoft Dynamics enterprise resource planning (ERP) and customer relationship management (CRM) solutions. With Microsoft Dynamics NAV implementations, you can use the Rapid Implementation Methodology

(RIM), part of Sure Step. RIM provides powerful tools that help automate Microsoft Dynamics NAV implementation tasks, including migrating master data.

For more information about Sure Step, visit https://mbs.microsoft.com/partnersource/newsevents/news/newsgeneral/introducingdynamicssurestep.htm?printpage=false.

Learn about RIM at https://mbs.microsoft.com/partnersource/deployment/methodology/projectmanagement/rapidimplementationmethodology.htm?printpage=false.

The MSF Team Model

One of the most important tasks in building a vertical or industry solution is setting up a team to manage the project. In building that team, be aware that consulting skills don't necessarily translate into effective product development. With the MSF Team Model, a team comprises a number of roles, and a single team member may hold a number of roles. The roles in the MSF Team Model cover multiple functional areas, including:

- **Product management**
  The primary goal of product management is to satisfy customers. This role covers four key functional areas: marketing, business value, customer advocacy, and product planning.

- **Program management**
  The focus of the program manager is to meet the goal of delivering the product within project parameters. This role covers four key functional areas: project management, solution architecture, process assurance, and administrative services.

- **Development**
  The development role focuses on "build to specification." This role covers four key functional areas: technology consulting, implementation architecture and design, application development, and infrastructure development.

- **Test**
  The goal of the test role is to approve the solution for release only after all product quality issues are identified and addressed. The test role covers three key functional areas: test planning, test engineering, and test reporting.

- **User experience**
  The goal of the user experience role is to enhance the solution's effectiveness for users. This role covers six functional areas: accessibility, internationalization, technical communication, training, usability, and graphic design.

- **Release management**
  The goal of release management is to manage deployment and ongoing operations. This role covers four key functional areas: infrastructure, support, operations, and commercial release management.

Key Quality Goal	MSF Team Role
Delivery within project parameters	Program Management
Delivery to product specifications	Development
Release after addressing all issues	Test
Deployment management and ongoing management	Release Management
Enhanced user performance	User Experience
Satisfied customers	Product Management

Table 1: MSF Team Model and Key Quality Goals (Source: "MSF 3.0 Overview" white paper)

For more information about establishing a team based on the MSF Team Model, read the white paper "MSF Team Model v3.1."Vertical Knowledge

When developing a solution for a vertical market or industry, team members should have an in-depth knowledge of the target industry.

Successful industry partners have developers who have an understanding of the functionality needed for the target industry. Developers with excellent knowledge of standard Microsoft Dynamics NAV can use those skills to assist program managers in designing the industry functionality.

For the product management, program management, test, and user experience roles, choose people with business process knowledge. You can select people from your existing consulting team, but separate out their work roles so that they have enough time to focus on the product development. Your vertical solution will be delayed if your staff is continually pulled back into specific customer implementations.

Consider working with outside resources to assist with feature specification, testing, and validation of the user experience. If you have customers working in the industry you want to target, consider using staff from those customers. Make sure, however, that the resources are from different customers; otherwise, your solution may not meet the general needs of the industry.

Consider people who have worked in the industry from three to five (or more) years, who participate in industry associations, and who perhaps have a relevant industry qualification. A review panel with a selection of customers and industry experts could help you with feature specification and selection.

The MSF Process Model

Every released product goes through a lifecycle process that includes all the activities from planning to completion and transition to an operational status. The main function of a lifecycle model is to establish the order in which you perform the project activities. The appropriate lifecycle model can streamline a project and help ensure that each step moves the project closer to successful completion.

A simple view of the MSF Process Model lifecycle is shown below.

For more information, read the "MSF Process Model V3.1" white paper.

Upgrade Considerations

Upgrading is an important part of scaling industry business solutions, so be sure upgrading is part of your planning process. If you design your industry solution so that it is easy to upgrade, you can use your resources more efficiently and have a higher maintenance percentage. For more information, refer to the "Best Practices for Upgrading to Microsoft Dynamics NAV" white paper, which discusses the roles and phases of upgrade projects using Sure Step and RIM (Rapid Implementation Methodology).

BUILDING INDUSTRY BUSINESS SUCCESS ON MICROSOFT DYNAMICS NAV

The Product Management Role

The primary goal of product management is to satisfy customers. Product managers are responsible for managing the project and guiding it to success. The following best practices will assist you in creating your product management team.

Business Value

Before you begin to build your solution, it is important to establish the business value. You can do this by asking your organization a series of questions.

1. *What is the business value of developing an industry solution around Microsoft Dynamics NAV?* To pursue an industryefficiently, you must have a commitment from management to build a solution that will satisfy the target market segment. For this process to be successful, business management needs to support product development.

2. *What industry are you going to pursue?* Management must be decisive in choosing the industry. This includes a proper analysis of the market and geographies to evaluate the size of the potential market. With this analysis, management can decide which markets will be profitable to pursue.

3. *How are you going to bring your solution to market?* Are you going to be the only company that works with your solution, or will you train and enable other partners to sell the solution?

The decision to develop an industry solution may be the result of an analysis of your existing customer base. If you decide to build an industry solution because you have been successful in selling to a particular industry and in making one-off customizations, then you must make an effort to separate the individual needs of the customer from the greater generic needs of the industry.

Business Model

In addition to the business value, carefully consider the business model. If you already have a successful consulting business built around Microsoft Dynamics NAV, you might establish a separate company or subsidiary for product development. In some countries, that approach can be an advantage because government funding is available to establish new product-development companies.

Establishing a separate company for development can also help if you want to expand your product development business. It's often easier to attract funding from investment or venture capital firms if product development is separated from your consulting business. In addition, such a separation allows you to raise capital with share offerings or other equity placements to fund product development.

Most important, your business model or consulting practice is an ideal knowledge base for your vertical solution. You can use your consultants to provide input on the features, requirements, and tests. However, because consulting and product development are different disciplines and require different skills, each should have a different business model to be successful.

Depending on your business model, you can choose to become an independent software vendor (ISV) that focuses on developing solutions that you resell to customers through a reseller channel. As an ISV, you may need access to PartnerSource and to the development and demonstration licenses and registration keys for Microsoft Dynamics NAV. You can sign the Microsoft Dynamics ISV Developer Registration Agreement (ISV DRA) and the associated ISV DRA Addendum to get the relevant access. Those documents can be found at www.microsoft.com/downloads/details.aspx?FamilyId=583A26FA-9562-4C42-B8CE-FE64BCF7CB3C&displaylang=en.

If you intend to resell the Microsoft Dynamics NAV base product, you should not sign the Microsoft Dynamics ISV DRA. Instead, sign the Microsoft Solution Provider Agreement (SPA), which grants you, as a reselling partner, the right to place orders with Microsoft Regional Operations Centers for specified Microsoft business-management solutions software licenses and related services. Ordering is contingent on having met SPA authorization requirements, which include passing proficiency exams on the software you plan to order from Microsoft. Find more information at
https://partner.microsoft.com/global/productssolutions/dynamics/solsprovideragreement.

Competencies

You should consider achieving two competencies in the Microsoft Partner Program (MSPP): the Microsoft Business Solutions (MBS) Competency and the ISV/Software Solutions Competency.

MBS Competency

The MBS Competency is a certification of resources. You can employ or contract with one or more individuals who, combined, must complete and pass three Microsoft exams and submit at least three customer references, each featuring the implementation, deployment, customization, or maintenance of Microsoft Dynamics NAV. When you enroll in the MBS Competency, you'll find a unique set of benefits and resources that align to your business focus and support all stages of your business cycle. The benefits you receive are in addition to those received at the Certified and Gold Certified Partner levels. For more information, see
https://partner.microsoft.com/global/program/competencies/businesssolutions.

ISV/Software Solutions Competency

The Microsoft ISV/Software Solutions Competency provides additional benefits and support to Microsoft Certified Partners and Gold Certified Partners who specialize in developing and marketing packaged software solutions based on Microsoft technologies. After finalizing the development of the add-on application solution, the developing partner can choose to have the solution tested. For more information, see
https://partner.microsoft.com/global/program/competencies/isvsolutions/40013116.

Partners achieve the ISV/Software Solutions Competency when they have a tested solution and three customers. Learn more at
https://partner.microsoft.com/global/program/competencies/isvsolutions.

For additional resources from the Microsoft Partner Program, review the materials in the Partner Vertical Resource Center at https://partner.microsoft.com/global/vertical.

Partnering

Partnering is a key element to succeeding in your chosen industry. You will build a number of relationships that include, but are not limited to, the following:

BUILDING INDUSTRY BUSINESS SUCCESS ON MICROSOFT DYNAMICS NAV

- **VAR partners**

  Existing Microsoft Business Solution partners who are resellers or value-added reseller (VAR) partners are key to the success of selling your solution. Most industry solutions, except those in heavily regulated industries, can be sold in other geographies. You can use tools provided in MSPP, like the Microsoft Partner Channel Builder, and Microsoft events, such as the World Wide Partner Conferences, to locate other partners. If you have reseller partners, you need to consider how they get your solution, how they are trained, and how you support them in the sales and post-sales processes. You could build your own channel to resell your solutions by working with IT solution providers who have knowledge in your industry. Also consider other product development organizations already working in the industry that have solutions nearing the end of their product lifecycle. Some successful Microsoft Dynamics NAV ISV partners have worked with organizations to offer new solutions to an existing customer base.

- **Training partners**

  If your industry solution has a potentially large market, you might consider working with training partners to educate resellers or potential customers about your industry solution. Establishing a relationship with training partners provides more flexibility and greater scalability. For example, you can provide training to larger audiences and you don't have to use your capital to set up professional training. There are Microsoft Certified Training Centers in all countries where Microsoft operates. You might also consider setting up a certification process to certify your channel partners.

- **Industry partners**

  Consider partnering with an industry association. Becoming an active member to help an industry cause and presenting at related conferences will help boost your credibility as a solution provider. Expert members can help you validate solution features, understand market trends, and identify directions in your target industry.

- **Localization partners**

  Bringing your solution to a new market can involve localization. You will need to ensure that your solution meets the regulations and requirements of the specific market. Consider working with a local partner to analyze local requirements so that you don't have to add additional resources to your team. This provides scalability because you are able to pursue markets faster than if you had built your internal team to support these markets.

- **Translation partners**

  Shipping your solution to different countries and geographies requires translating the user interface to the language requirements of that country. It is unlikely that you will be able to provide internal resources to fulfill the translation needs of every country. Engage an established professional language translation service so that you can take your product to different markets quickly.

Managing Risk

A development project involves a number of risks. Although managing risk is not addressed in detail in this white paper, it is an important part of the process and methodology. Here are a few points to consider:

- **People**

  People are a primary resource and the most important part of any project. The best way to avoid people-related risks is to make sure you have excellent documentation

practices for all phases of product development. Such practices reduce errors and ensure compliance with project processes.

- **Hardware and software**

  You are developing software that is residing on hardware. Therefore, make sure you have excellent back-up and recovery procedures for your source control, development databases, released code, service packs, hotfixes, and all other materials associated with the development project. Government regulations, such as Sarbanes-Oxley, determine how long you must keep this material, but, as a minimum, keep it long enough to support the published support and release cycles.

Marketing

The sales and marketing teams must be involved in the development process to provide input into product features and packaging in order to target the right market. The marketing team should assist in selecting target release dates to suit the buying patterns of your target industry.

Branding and Market Differentiation

The key role of the marketing team is to brand your solution. Branding gives you the opportunity to differentiate your business from that of competitors. Ensure that your solution meets the needs of your customers so that they don't focus on price as the primary differentiator when deciding whether to purchase from you or your competitors.

How do you differentiate your business from other businesses in the market? If you participate in a co-marketing event with Microsoft, such as the annual Microsoft Dynamics Convergence conference, other Microsoft Dynamics NAV resellers will be present. You can get customers to remember your solution when the conference is over by setting it apart with your unique branding. In addition, customers know your solutions are built on Microsoft technology, which gives them confidence they are purchasing an industry-standard solution.

Certified for Microsoft Dynamics

The Certified for Microsoft Dynamics standard is another way to help your solutions stand out. Certification is designed to help existing partners with packaged software solutions that have been in the market for some time to stand out as meeting the highest Microsoft Dynamics standard for partner-developed software solutions. For solutions that become certified, partners receive a distinctive logo to use with their marketing materials and become eligible for additional marketing benefits. Certified for Microsoft Dynamics solutions are developed to connect, integrate, or operate with Microsoft Dynamics business solutions, including:

- Microsoft Dynamics AX

- Microsoft Dynamics GP

- Microsoft Dynamics NAV

- Microsoft Dynamics SL

- Microsoft Dynamics CRM

The Certified for Microsoft Dynamics standard represents an opportunity to accelerate growth and market reach. Microsoft encourages partners new to Microsoft Dynamics to work toward certification. Find more information on Certified for Microsoft Dynamics at www.InnovateOnMicrosoftDynamics.com/cfmd.

Vertical or industry focus

By focusing on a vertical market/industry, you can simplify efforts for your marketing team. That focus enables them to produce directed marketing materials and allows you to work with better-

qualified prospects, thereby reducing sales efforts and overall marketing costs. You can also establish relationships with prospects in your target industry and use CRM systems to locate and track their information and possible investment plans. When you have an in-depth understanding of your industry, you can understand the industry cycles and know when to ramp up selling efforts in the purchasing cycle.

In addition, you can use your industry associations to brand and improve your image. Speak at and sponsor events with other industry organizations so that others will associate your industry solution with your organization. Before you release the solution, marketing will prepare launch events and associated activities. Because the marketing team works with long cycles, activities such as booking venues and planning public relations events need to be done well in advance. Missing event deadlines and milestones can result in financial losses, so everyone involved must be committed to meeting their dates.

The Microsoft Partner Program and Microsoft Dynamics Vertical /Industry Marketing initiatives provide marketing materials and tools at the following Web sites:

Microsoft Partner Program
https://partner.microsoft.com/global/salesmarketingsection/smcampaigns
https://partner.microsoft.com/global/partner

Microsoft Business Solutions Industry Portal
www.microsoft.com/dynamics/industry/default.mspx

Microsoft Partner Vertical Resource Center
https://partner.microsoft.com/global/vertical.

Partner Solution Profiler
https://partner.microsoft.com/global/solutionprofiler

Presales Information

After your solution is developed, provide your sales team or channel partners with the tools and information they need, which include, but are not limited to, the following:

- **Boxed product**

  If you have planned well and provided installation tools with well-documented implementation procedures, you can consider packaging your solution as a boxed product. This works well when you expect little or no end-customer customizations and you want to develop a volume solution for a market segment. If you plan not to have end-customer customizations, then you can easily ship upgrade packages to migrate your customers to new versions of your solution. You can also have variations for different subsegments in an industry.

- **Brochures**

  Brochures are a valuable form of describing your solution to prospective customers. Brochures can be used for direct marketing, events, and RFP responses. The brochures should include information such as key customers, case studies, and industry certifications.

- **Data and fact sheets**

  A data sheet is a brief summary of the features in your solution. Consider producing data sheets that compare your solution to competing solutions, and include customer case studies to illustrate the benefits.

- **Presales demo material**

  How prepared is your sales team? Preparing demo scripts that describe how to present and demonstrate your solution can mean the difference between winning and losing a sale. You can create workflow demo scripts with Microsoft Visio® software or create a presentation using Microsoft PowerPoint® software. You can also create demo videos

using tools such as Microsoft Windows® Movie Maker and Techsmith [Camtasia Studio software](#).

- **White papers**

  White papers provide information about the solution and its features in greater detail. In the sales process, you can provide white papers to those evaluating the technical details. Be sure to instruct sales teams about when to use technical presales materials so that features and competitor information does not distract from the sales process. Focus on the features of your solution that solve your customers' business problems so that customers understand the value.

- **User communication**

  How are you going to communicate the availability of your product to customers? Seminars and industry events may be good venues in which to present and showcase new releases. It's also important to have a strong Web presence that includes compelling elements such as online demos.

- **Case studies**

  Customer evidence is one of your most important selling tools. Because prospective customers don't always like to the first to purchase a new product, use a customer case study to show prospective customers that you have been successful at implementing your product. Of course, this may seem like an impossible situation: You can't get a customer if you don't have a case study, and you can't get a case study if you don't have a customer. Your first customer will always be the most difficult to get and the one you want to keep. For case-study candidates, consider the customers you used in your initial analysis studies. Or, create a pilot site and negotiate with the customer, or absorb the costs of a customer's implementation so that you can get the reference.

  You can also work with Microsoft after you have developed your case study. If the solution uses an interesting combination of Microsoft technologies, there may be various ways to have your case study advertised through Microsoft programs.

- **Industry certifications**

  In marketing your solution, demonstrating its quality is important, and industry certifications are a quality stamp. We recommend trying to get your solution on preferred industry purchasing lists, which may require joining or becoming certified by an industry association. If you are working through a partner channel, having marketing materials ready can mean the difference between a partner working with you or not. The materials show that you are committed to the success of the solution.

Customer Advocate

The product management team is the link between your company and your customers. Communicate to customers the lifecycle of your product, get them excited about upcoming releases, and collect their feedback. Then, you can use your support system, covered later in this paper, to monitor customer satisfaction and ongoing needs.

Being a customer advocate can help you integrate customer needs during the product development process. Consider incorporating review feedback after support cases are closed. This allows customers to submit new features. Remember to keep customers updated on the status of support cases and the acceptance of submitted feature requests.

Acting as a customer advocate can help integrate customer needs with the product development process. By building an industry solution, you can create a more focused user community for that solution. Consider setting up a user group, a concept that works well when the members share a common interest. Customers are interested in running their business smoothly, and your solution provides the tools they need to manage their business more

effectively. By having a user group for your industry solution, members are able to network and understand how others are solving similar business issues.

Product Planning

Product planning involves the overall management of your solution. The planning process can start only after the business model has been developed. Product planning involves understanding the business needs of the industry. From the business needs analysis, the product planner can create a feature set to be included in the product release.

During planning, you will need to include a number of deliverables to produce a packaged product, including, but not limited to:

- Business requirements

- Documentation

- Training materials

- Implementation materials

- Sales and marketing materials

Short-term, mid-term, and long-term development plans should be an integral part of product planning. During development, the focus should be on what kind of functionality you want. Part of your planning process should consider what features are required for different marketing segments, which allows you to target your marketing and sales efforts.

Business Requirements

The business requirements analysis is one of the most important parts of the product lifecycle. In this stage, you can map customer needs to your solution. Those customer needs will be basis for the features you add to your solution. You can use various resources to assist with market research, including:

- **Your team**

  Use the knowledge of the resources you have, such as the consultants and the management team in your organization, and survey your organization about their past work experience. Also consider recruiting additional subject matter experts from other companies; for example, competitors or experts in the specific industry, such as a production expert from the clothing manufacturing industry. This experience— combined with other market research—can give you the initial information you need to develop your industry solution.

- **Your customers**

  Look to your current customers for business requirements. Your current customers are a great place to start because working with them may also provide you with case studies to use in marketing your solution. Make sure to use various customers and combine this information with other research. Ask customers if they are having difficulty automating other parts of their business. Perhaps you'll discover an opportunity to facilitate cross-industry trading that leads to a second industry solution, for example, pharmaceutical companies and pharmacy retailers. Or, there might be a way to provide an vertical solution to assist the interaction of these two business types in the supply chain.

- **Industry experts**

  Look to industry experts for guidance on industry trends. Perhaps a government regulation is about to change, mandating additional processes in an industry. Use industry experts as consultants to gain insight and learn how the industry works. If the

experts have a good reputation, you can also work with them in sales and marketing to endorse your solution.

- **Research**

  Research is the most expensive way to obtain or validate requirements, but it can be the most rewarding because you may discover an untapped market. Start with the industry classification, such as SIC (Standard Industrial Classification) or NACE (Nomenclature d'Activités de la Communauté Européenne) codes, and then narrow down a set of industries to investigate based on market size. Investigate the software that is being used in an industry by setting up a market study and interviewing customers. Those types of research can illuminate whether existing software is outdated or inadequate, which can lead to new opportunities. Also look to government economic data and census data.

- **Industry associations**

  Look to industry associations for trends. Industry associations may have endorsements; for example, members may only purchase products endorsed by the association. If the association is able to certify your solution, make sure you plan to have your solution certified. Industry associations enable you to network and provide a place to market your solution. You should also look to becoming an active member of the industry association by sponsoring events. If the association requires your staff to become certified, as, for example, the American Production and Inventory Control Society (APICS) does, then work on certification. Having an industry-certified staff can be an added sales feature for your solution.

- **Pilot studies**

  Pilot studies involve developing a set of features into a prototype and implementing the prototype at a test customer site. The advantage of pilot studies is that you are able to gather feedback from the customer on feature usability and feature gaps. However, it is important that you don't add features that suit only the pilot customer, so conduct further research to be sure the features are needed for the industry you are targeting.

- **Competitive research**

  Conduct a competitive analysis of your solution during the planning phase. This helps ensure your solution has the features to compete with other solutions on the market. If you don't offer more features than the competition does, then make sure you are offering a new usability experience that is an improvement compared to that of the competition. By using the Microsoft platform and its technologies to build a strong value proposition, you can compete with other solutions based on older Unix-based or legacy platforms. By being able to use information in different ways, in comparison to a legacy platform, your customers can achieve a fast return on their investment.

Obtain information from a number of sources so that you have a balanced opinion. For example, if you limit research to your customer base, then you could miss features that would allow wider distribution for your solution.

Release Cycles

Set up a consistent project plan—that management has agreed on—regarding when you will release solutions. A consistent project plan enables you to manage your development cycles appropriately and communicate consistently to your channel partners and customers. Management and other parties need visibility into release cycles to help manage their organizations.

The Microsoft Dynamics NAV shipping cycle is approximately 18 months for a new version, with a service pack release approximately every six months. Releasing a new version of your solution on the same cycle allows you to be involved in Microsoft launch events and other

campaigns. By incorporating new releases of Microsoft Dynamics NAV, you can consistently encourage your customers to upgrade.

It's advisable to create a service pack for your product to be released after the Microsoft Dynamics NAV service pack is released. That helps minimize the releases you need to ship to the customer and your channel partners, and helps you align consistently with Microsoft. As a guideline, plan to release your product three months after Microsoft releases a version. As for service packs, it's best to establish a plan that fits your industry functionality.

Know the Target Markets

In the business plan, you will have prioritized which countries or geographies to target. Some markets have bigger potential than others. The product planner will have to calculate release schedules that are aligned with your company's goals. Keep in mind that locations will have different localization and language support requirements. As a result, you will have to plan your resources so that you get the best market coverage in the least amount of time. You should work with management to prioritize the schedule.

Introducing your solution into new markets is one of the best ways to expand your market presence and increase revenues. But finding the right tools, guidance, and partners to enter new markets effectively can be costly and time-consuming. Microsoft can assist you in building a high-performing channel and provides several resources: ISV Channel Program Workbook, International ISV Assistance Portal, and ISV Channel Development Service, all described in more detail at https://partner.microsoft.com/global/program/competencies/40045529 .

The Program Management Role

The focus of the program management role is to meet the goal of delivering the solution within project constraints.

Project Management

The project management team is responsible for driving the calendar. They need to keep the project on time at every stage of the development cycle. Project management also needs to help keep morale high in the project team.

You will need a strong project management team *or* person, depending on the size of your solution. That team or person will have a number of tasks that include, but are not limited to:

- **Developing the project schedule**

  All team members need to provide input and be committed to the schedule. The schedule is the tool for communicating progress to business management, such as marketing, channel partners, and customers. The schedule should be realistic and align with the release cycle management has agreed upon. The project management team should also create project milestones and celebrate with the project team when these milestones are reached, or use them to bring the project back on track. Project management works with product management to control features. For example, to meet a committed market date, the product team may need to drop some features in a release.

- **Managing resources**

  The project manager should be allocated a development budget that covers the entire development phase. The project manager then has control in allocating the right resources at the right time. The project manager should update business management on the state of the project with status reporting.

- **Communication**

  Communication is a key element in successful project management. The project manager has control over the schedule, and it is his or her responsibility to hold regular status and update meetings. The project manager should also work closely with the HR representatives in your organization to help identify personnel issues that could result in the project not being delivered, as well as drive team morale by fostering open communication and taking input from all members of the project team.

Solution Architecture

A senior developer works with product management to design the overall architecture of your solution, providing guidance on the data model, integration points, and functionality to include in the product. Typically, the developer's job title is solution architect. Ideally, the solution architect will have some experience in the industry.

The solution architect manages the overall functional specification and solution design with input from the development team. The solution architect undertakes a number of tasks, including:

- Defining terminology for the product development

- Designing the data model in-line with needed industry functionality

- Designing and specifying the technology platform to use as well as when to make a technology switch that enables greater features at reduced development costs

- Designing the integration points

- Working with the functional scope at feature reviews

- Defining security

- Selecting development tools

Terminology

Consistent naming conventions are important for controlling the quality of your product. Microsoft Dynamics NAV provides a number of guidelines to help you understand terminology and naming conventions.

- **C/AL Programming Guide**

  The *C/AL Programming Guide* is located on the Microsoft Dynamics NAV Tools CD. Make sure that all your developers read and adhere to the guidelines outlined in Chapter 2, "Naming Conventions," and Chapter 3, "Number Conventions."

- **Application Designer's Guide**

  The *Application Designer's Guide* is located on the Microsoft Dynamics NAV Product CD. All of your developers should read this comprehensive document.

- **Terminology Handbook**

  The *Terminology Handbook*, located on the Microsoft Dynamics NAV Product CD, is an excellent guide for terminology about all objects in the C/SIDE® development environment. It also provides additional information on name objects in Chapter 4. You'll find a "Terminology Review Checklist" in Section 7.4.

- **GUI guidelines**

  Prior to version 4.0, Microsoft Dynamics NAV maintained strict GUI guidelines as support for developers. The guidelines document was not updated for 4.0, but the content remains valid. The GUI guidelines are located on the Microsoft Dynamics NAV Tools CD. All developers should review the guide. This is an excellent tool for understanding the strict standards inside the Microsoft Dynamics NAV application. The benefit of using the GUI guidelines is that you can check them with the GUI tool provided on the Microsoft Dynamics NAV Tools CD. Be aware that this tool is consistent with the GUI guidelines prior to Microsoft Navision 4.0, so the standard Microsoft Dynamics NAV code that has new user interface features will show up in the tool.

- **Industry terminology**

  You should also consider the industry language that you use for your solution. This may mean that you translate "Customer" to "Client" for a professional services–based product. In addition, you should create a terminology list for your solution. For more information on the language tools provided with Microsoft Dynamics NAV, refer to the section on the user experience role in this paper. Balance any changes in terminology with being able to offer a generic product and minimizing upgrade issues.

Simplicity

You should consider simplicity of design as an important way to approach development. The terminology you choose can add or subtract from a product's simplicity. If you are presenting the application with terminology specific to the industry, then your users are more likely to find their way through the application. Also, make sure you develop a menu system that reduces complexity. Use the role-based conventions introduced in Microsoft Dynamics NAV 4.0.

Consider the balance between maintaining changes to the standard Microsoft Dynamics NAV application and adding new functionality. When designing your additional industry functionality, design the integration with Microsoft Dynamics NAV in a way that minimizes maintenance. For more information, refer to the following section on the development role.

Prototyping

Microsoft Dynamics NAV is a great tool for prototyping. In the design phase, you can use prototyping to create a mockup of your design and use the prototype in a pilot project to provide pilot customers with a usable system so that they can envision the complete working system.

Developers can use the standard Microsoft Dynamics NAV application to create a prototype design for new functionality. Use this design in review meetings so that product management and developers can agree on the features to be developed. If you create a prototype, make sure that you don't develop from a prototype. Use the prototype to test concepts the solution architect is considering and to review features with test customers.

Reverse Engineering

Reverse engineering is a process of trying to understand the existing application. Part of the role of the solution architect is to understand where to make changes to the standard Microsoft Dynamics NAV application. Knowing where to make changes comes from experience in understanding the standard Microsoft Dynamics NAV application.

With the Microsoft Dynamics NAV Developer's Toolkit, located on the Microsoft Dynamics NAV Tools CD, you can use the Source Analyzer to reverse-engineer the code in Microsoft Dynamics NAV. With Source Analyzer, you can:

- Provide an object tree to navigate the C/SIDE objects.
- Provide an object diagram tool.
- Understand the relationship to tables.
- Understand the relationship from objects.
- Understand where objects are used.
- Search through the objects with Source Finder.
- View code behind the objects with Code Viewer.
- Obtain an overview of the objects used in a sequence of code with Method Flow.

You should also read the *Application Designer's Guide*, located on the Microsoft Dynamics NAV Product CD. This guide has several chapters that describe the object model in Microsoft Dynamics NAV. It also includes a flow chart of the object triggers in the various objects.

With the Source Analyzer and access to the business logic code in Microsoft Dynamics NAV, you can reverse-engineer Microsoft Dynamics NAV so that you can decide where to position changes.

Microsoft Dynamics NAV Patterns

The standard Microsoft Dynamics NAV application conforms to a number of patterns that you can follow when you are developing your new industry functionality. It is important to understand those patterns when making changes to the existing Microsoft Dynamics NAV application.

The *C/AL Programming Guide* and the *Application Designer's Guide* provide an overview of a number of the patterns. You can find additional information in the *C/SIDE Reference Guide*, which can be opened from the Help menu of the Microsoft Dynamics NAV Client. Consider the following topics in your solution architecture:

- **Multilanguage system**—How to work with the multilanguage system in Microsoft Dynamics NAV.
- **Code performance**—Excellent material on developing C/AL code for performance. Additional material is provided in *the Performance Troubleshooting Guide* and the

Microsoft SQL Server® Resource Kit, which are included on the Microsoft Dynamics NAV Tools CD.

- **Number system**—This ystem to autogenerate business document numbers.

- **Dimension system**

- **Sum Index Flow Technology (SIFT)**—In Microsoft Dynamics NAV, provides a unique way to provide calculated balances to users.

- **Batch reports**—To automate a data maintenance operation consider using a Codeunit and a form for future upgradeability.

- **Object numbering**

- **Naming conventions**

- **Ledger tables**—All submodules in Microsoft Dynamics NAV have a ledger table, which is the primary accounting table for that module. Never write to a ledger table from any interface, even C/AL code. Always process data into a journal, and then use the relevant posting Codeunits for that journal to post through to the ledger.

- **Journal posting Codeunits.** Section 3.2 of the *C/AL Programming Guide* describes some of the patterns you will see for journals. For example, General Ledger Journal posting is handled by Codeunit 11, 12 and 13. You will also see the same pattern in Item Journal posting, which is implemented in Codeunits 21, 22, and 23.

- **Invoice posting.** Section 3.2 of the *C/AL Programming Guide* describes some of the patterns you will see in code. For example, Sales-Post is Codeunit 80, and Purch-Post is Codeunit 90. They have associated Yes/No and Post+Print Codeunits 81, 82 and 91, 92, respectively.

- **Temporary tables**

- **Locking and concurrency patterns**

User Communication and Feedback

It is vital that your product management team and solution architect have a clear vision and goals. Conduct user-feedback sessions to validate the design of your industry solution. Overall, the solution architect will have the final say over what features make it into the product, but consider the following groups of people for gathering input:

- Customers
  As mentioned before, customers typically have a good idea of their specific needs, but they may not know the broad needs of an industry industry. Customers and industry experts almost always tend toward a more inflexible feature set, that is, a set closer to a system with which they are familiar. The challenge is that the shortcomings of that feature set is likely what you were trying to solve with your industry solution in the first place. Customers are good judges of functionality, but often poor judges of how to best build a solution with tools to enable this functionality. Therefore, customer input has to be carefully evaluated by program management.

- Industry experts

- Existing partners

- Sales and marketing

Integrating with Other Applications

When developing your industry solution, you may need to move the data into other applications so that users can manage in different ways and in various parts of the business process. Prebuilt and seamless integrations with other lines of business applications and with

applications like the 2007 Microsoft Office system are valuable to customers and can help improve sales. Several tools are available for developing integrations into Microsoft Dynamics NAV, including:

- **N/ODBC.** You can use this open database connectivity (ODBC)–compliant data interface on the native Navision database. With Microsoft SQL Server®, use the standard Microsoft drivers. ODBC is useful if you need to provide data access to other applications, such as report tools like SQL Server Reporting Services. You can't access the Navision business logic with ODBC drivers.

- **C/FRONT.** C/FRONT is a traditional programming Application Programming Interface (API). You can find documentation on how to use C/FRONT in the System Development Kit (SDK) documentation provided on the Microsoft Dynamics NAV Product CD. C/FRONT is an API that is implemented as a DLL file and an OCX that you can use in traditional programming languages such as C++ and Visual Basic. In Service Pack 1 for Microsoft Dynamics NAV 4.0, C/FRONT is provided as a .NET assembly that you can use to develop integration with any .NET-based programming language.

  C/FRONT is an efficient API in Microsoft Dynamics NAV table operations. However, it can't access the business logic in Microsoft Dynamics NAV. For example, you can interface at a data level, but can't trigger posting code. You should use C/FRONT only if you have an excellent understanding of the design of the Microsoft Dynamics NAV tables. You can create staging tables in your industry solution and populate data into those tables with C/FRONT. Then create a C/AL Codeunit that reads the data from the tables and posts the data. You must use the Microsoft Dynamics NAV Client of the Microsoft Dynamics NAV Application Server (NAS) to process the code. That can be done with the Job Scheduler in the Microsoft Dynamics NAV Client, for example, or in a looping or event-driven trigger in the NAS.

- **Microsoft Dynamics NAV Application Server (NAS).** NAS provides a way to run server-based processes that can execute business logic. Data can be received over a number of different transport mechanisms like Microsoft Message Queuing (MSMQ) and TCP/IP. You will have to create your own messaging system from your sending application and then interpret and process the messages received by the code triggered in Codeunit 1 trigger 99 in Microsoft Dynamics NAV. You can use the NAS to integrate to an application with that messaging approach.

  In order to save development time, you can purchase a framework, available from some Microsoft Dynamics NAV ISVs. One such ISV is the Canadian-based partner Digital Vantage Point, a participant in the development of this paper. You can use the FieldRef and RecordRef objects in Microsoft Dynamics NAV to build generic Codeunits to refer to any table in Microsoft Dynamics NAV. Microsoft Dynamics NAV uses the NAS as the integration technology in the Microsoft Dynamics NAV Employee Portal (SP1), Microsoft Dynamics NAV Commerce Portal, and Microsoft Dynamics NAV Commerce Gateway modules. Partners have used the NAS to integrate with Web sites and other line-of-business applications. Combined with Microsoft BizTalk® Server, NAS can be used to build real-time integration with almost any other application. You can read more about developing Microsoft Dynamics NAV Application Server code in the SDK documentation provided on the Microsoft Dynamics NAV Product CD.

- **COM.** You can extend Microsoft Dynamics NAV C/AL code with the use of COM (component object module). Microsoft Dynamics NAV C/AL variables can reference COM objects registered on the workstation. Use COM to integrate with Microsoft Office applications such as Microsoft Office Word and Microsoft Office Excel®. You can see this code implemented in a number of places already in the Microsoft Dynamics NAV application, such as the General Ledger Dimension Analysis Export to Excel functionality. The use of COM is also a common way to work with XML by interfacing with the Microsoft XML Parser. Because you can call COM in C/AL code, you can also

get the Microsoft Dynamics NAV Application Server to execute this code to automate a server-based integration with COM. COM can also be used to interface with devices such as serial printers. This approach is used successfully in the retail application provided by Landsteinar Strengur and also by Tegos, both participants in the development of this paper, in their waste management solution to interface with scale systems.

- **Custom controls**. Currently, the C/SIDE environment does not provide the ability to create new visual controls or host ActiveX® controls. You can work around this limitation with the use of COM. If you develop a new visual control in legacy languages, like Visual Basic 6.0 or new WinFormsbased controls running on the .NET framework, you can provide additional user interface features by calling these controls with C/AL code through COM references. You'll find this functionality implemented in the standard Microsoft Dynamics NAV application with the manufacturing Gantt chart control.

- **Dataports and XMLPorts.** You can use Dataports to migrate data in traditional CSV (comma-separated variables) and flat file formats. Use XMLports to provide similar functionality to Dataports but read and write using XML files.

- **Staging Tables**. Some ISVs provide staging tables for data. That approach is commonly used on SQL Server–based installations. In this way, you can use standard SQL data operations and services, such as Data Transformation Services, to integrate data from other applications. This has a performance advantage, as all the execution runs on the database. But, you can't execute business logic, so you will have to combine this approach to integrate with a Microsoft Dynamics NAV Client or the Microsoft Dynamics NAV Application Server that is configured to sweep the staging tables and post to Microsoft Dynamics NAV on a periodic basis. Microsoft Dynamics NAV has a feature that enables you to create a C/SIDE table that is linked to a SQL view, which can help in these types of integrations. You can read more about this in the Microsoft Dynamics NAV *Application Designer's Guide*.

- **Web services.** To consume a Web service, you can use COM components like the Microsoft XML Parser to do HTTP POST after forming a SOAP message. You can also use Microsoft Visual Studio® .NET to create an ASP.NET Web service. The implementation of the Web service would use C/FRONT or the Microsoft Dynamics NAV Application Server to extract and interact with Microsoft Dynamics NAV.

If you do reference and use external COM components to add functionality to your industry solution, be sure you understand the implications by asking questions such as:

- Do you have a license to use this control?

- Is it available on the target platform? Does this limit the use of Microsoft Dynamics NAV on specified operating systems?

- How do you deploy this control?

- How do you compile code? Developers will need a copy of the control on their computer so that a full compile can be completed.

Security

Security is of vital importance to Microsoft and its customers. Be sure your solution is developed and implemented using industry-standard security technology. An overview of the Microsoft Dynamics NAV Security Model appears in Chapter 3 of the *Application Designer's Guide*. You are encouraged to read the *Microsoft Dynamics NAV Security Hardening Guide*, which is available on the Microsoft Dynamics NAV Tools CD.

When designing your industry solution, consider securing the data in your application by using firewalls and encryption, which involves building new security roles or modifying existing Microsoft Dynamics NAV security roles.

The Microsoft Dynamics NAV Tools CD has a guide and a set of tools to help you assign permissions to roles. You can use the tools to create demo data for your solution. You can also use them to generate roles for testing functionality.

For general background information and to stay current on issues regarding security in software development, visit the MSDN Security Development Center on a regular basis to review the security documentation at: www.msdn.microsoft.com/security/.

For more background information, read *Writing Secure Code* (Second Edition) by Michael Howard and David LeBlanc at: http://www.microsoft.com/MSPress/books/5957.asp. The safety of your development code is another important consideration. Control who has access to the code in the development cycle, and make sure changes are controlled. A source control tool will help you manage your code.

Developing an external control may have some upgrade benefits, as you would be linking this functionality into the existing C/SIDE environment with C/AL code. This is easy to upgrade, and it may mean that the control doesn't have to change in a new version. You must also consider controlling the security of the interface, as forms developed in C/SIDE will enhance the security with the use of the application security provided in Microsoft Dynamics NAV. If you develop externally in .NET, then you can make use of authentication provided by the Microsoft Windows® operating system.

Process Assurance

The program manager's responsibility is to drive the quality of product development, including by defining the benchmarks and goals the developers and testers should be working toward. Various tools can help you control quality and manage the project schedule, including:

- **Microsoft Office Project.** Use this application to control resources and manage the project timeline. You can use this as input into your project status. A number of ISVs interviewed for this paper have built a system in Microsoft Dynamics NAV for this purpose. You could also use a simple tool like Microsoft Office Excel® to manage your project.

- **Source Control (SC)** – Use source control to track the progress of developers. This system gives you some overview statistics from SC, for example, how many objects a developer is working on and the history of development. You can also use SC to branch your project, a good idea if you want to eliminate features for a particular release.

- **Support System** – Use this to gather metrics to track your project after the release of your solution and to gather customer feedback to help you plan the next product release.

- **Bug "Defects" Tracking System** – Use this tool to get feedback on the quality of your product development. The product testers will create defect items in the tracking system after their testing cycles, and the defect tracking can be used to control the project. You should have a regular cycle of checking the defects, perhaps on a daily basis, or at least after the testing cycle. Send the defect report out to all the team members. If defect counts are not being controlled, then consider stopping new feature development in favor of eliminating defects. You should be aiming for a zero-defect release, so prioritize eliminating defects over releasing a full feature-rich solution. If you decide to eliminate features to get a project on track, then make sure that is clearly communicated to product management.

- **Microsoft Office SharePoint Server.** Consider using Microsoft Office SharePoint® Server to coordinate all project documentation. You could also use the new team tools provided in Microsoft Visual Studio® 2005. Visual Studio 2005 has an integrated team system that allows you to set up a team project. You can use the project together with the source control system, manage documents with Office SharePoint Server, and

manage work items as well as bugs or "defects." You can read more about the features at the Visual Studio Developer Center at http://msdn.microsoft.com/vstudio/.

Administrator Services

Program management looks after the administrator side of the development project. You must take Microsoft Dynamics NAV licenses into account if you are developing a product that will be registered with the use of the license system. Register your development license with Microsoft in order to update it with the ability to develop objects in a number range reserved for your solution. You receive the object number range, unique for your organization, after you register with Microsoft. For more information, visit PartnerSource at https://mbs.microsoft.com/partnersource/deployment/documentation/howtoarticles/NAV_AddOn Registration.htm

To register your solution, you must submit the Add-on Registration Form, which must be processed in the region(s) where you plan to sell your solution. Start the process well before you intend to go to market so allow enough time for the Microsoft licensing systems to be updated.

When working with reselling partners or end customers, clearly communicate who the solution developer is. Other Microsoft partners and customers should not be given the impression that Microsoft Dynamics has authorized or certified your solution. Only Certified for Microsoft Dynamics solutions can claim that status. Customers must understand that Microsoft Dynamics does not control the development of your solution and does not take any responsibility for supporting or upgrading either your customizations or your solutions.

The Development Role

The focus for the development role during a Microsoft Solutions Framework project is the build-to-specification goal. It is expected you have a good understanding of the C/SIDE development environment and the C/AL language.

If you are new to developing with Microsoft Dynamics NAV, use the following recommended tools and resources:

- **Training resources.** Access Microsoft Dynamics NAV training information at www.microsoft.com/dynamics/using/training.mspx.

- **Microsoft Dynamics Community Web site**. Keep up to date with fellow Microsoft Dynamics NAV technical people at http:www.microsoft.com/dynamics/nav/community.mspx.

In addition, the following is recommended background reading for Microsoft Dynamics NAV:

- *Microsoft Dynamics NAV Application Designer's Guide*, available on the Microsoft Dynamics NAV Product CD

- *C/AL Programming Guide*, available on the Microsoft Dynamics NAV Tools CD

- *GUI Guidelines,* available on the Microsoft Dynamics NAV Tools CD

- *C/SIDE Reference Guide*, installed with the Microsoft Dynamics NAV Client

We recommend that a Microsoft Dynamics NAV developer attend the Microsoft Dynamics NAV developer training. Also consider attending the training for functional modules on which your industry solution will be based.

The documentation and training give the Microsoft Dynamics NAV developer the necessary background information for using and working with the C/SIDE development environment. The aim of the following sections is to point out information that is complementary to the existing development material.

Technology Consulting, Implementation Architecture, and Design

The development of design and functional specifications requires a deep understanding and knowledge of the following environments and technologies:

- Microsoft Dynamics NAV C/SIDE environment

- Microsoft Dynamics NAV Application

- Microsoft development tools, such as Visual Studio and the Microsoft .NET Framework

- Microsoft applications, such as Microsoft Office and Microsoft BizTalk Server

- Operating system technology, such as Microsoft Windows Server® 2003.

The developer role uses that deep knowledge and understanding to assist program management in defining functionality and choosing technologies. Technology choices are in the following areas:

- **User interface design.** *What role in an organization does the functionality serve?* If the user is a back-office user, then developing C/SIDE forms running in the Microsoft Dynamics NAV Client would be excellent for rapid data entry needs. *Will the internal user need to access data from Microsoft Dynamics NAV on a casual basis?* Consider developing extensions to Microsoft Dynamics NAV Employee Portal help. *Is the user external to the organization and working remotely?* Consider developing a new .NET Smart Client application by using standard Microsoft development tools such as Windows Forms (WinForms) or Web-based tools with ASP.NET.

- **Integration with the existing application.** If you are an ISV and want to develop integration with an existing industry standard application—for example, a scales system commonly used in the meat industry—*how should that integration be built?* The developer looks at the integration options for Microsoft Dynamics NAV and helps choose the best one for the job.

- **Development tools.** *What development tools should be used?* The choices will vary depending on the type of functionality that needs to be developed. C/SIDE provides the developer with a robust, simple environment that abstracts a lot of the low-level development tasks so that the developer can focus on coding the business logic that provides a solution for a customer. However, the code is executed by the Microsoft Dynamics NAV Client. When developing new functionality for some roles, it may not be possible to use the Microsoft Dynamics NAV Client; therefore, functionality could be developed externally and integrated to Microsoft Dynamics NAV. The .NET Framework provides an excellent set of tools and environment to build an application for any need, such as a mobile device–based application.

- **Standards-setting.** Because the developer will have to execute on any development standards, the developer should contribute in setting those standards, including defining internal development standards such as naming conventions, best practices, code reviews, statistics, and tools to help measure the development effort.

Application Development

It is up to the developer to take the functional specifications developed by program management and turn them into a product.

Setting Up a Local Development Machine

A new local development machine is important for consistency among the developers on a team. Use the following guidelines to set up a machine for developers:

- **Install the development tools.** For a C/SIDE developer, this involves installing a Microsoft Dynamics NAV Client. Because of language issues, you should develop with a W1 client of a client that uses an ENU fin.stx file. The client will need a copy of a developer's license. They may also need to install the Microsoft Dynamics NAV Developers Toolkit so that they can use some of the features, such as Source Analysis. You may also need other third-party development tools, for example, a popular code comparison tool such as Araxis Merge.

- **Create an account in the relevant systems**, such as Source Code control, Support System, and Team Shares.

- **Set the regional settings to the development language.** When developing in C/SIDE, we recommend developing in U.S. English using a decimal separator set to "." (period). This setting can be viewed and modified in the Regional Options applet in your system's Control Panel. C/SIDE properties that expect a decimal number, such as InitValue, may not be interpreted correctly if the wrong decimal separator is used.

- **Recommendation:** Develop in U.S. English. and then use the multilanguage tools to translate the application to the target language.

Developing with C/SIDE Objects

The Microsoft Dynamics NAV *Application Designer's Guide* is the best place to get an introduction to developing and working with the objects in C/SIDE. The following information is provided on some of the objects to help you develop your industry functionality.

Tables

Modifying the data model is one of the main tasks you will undertake to implement your industry functionality. You do this by adding new tables in your add-on number range or by modifying existing tables in the number range below 50,000. Some of the best practices ISVs use to minimize upgrade and migration work and simplify development include:

- **Recommendation: Do not copy existing tables to implement new functionality.** If you do, you will have to modify forms, reports, and Codeunits that reference this table to make sure that the application continues to function.

- **Consider normalizing the data model for additional information for your industry solution, and storing data in an additional table.** That approach allows you to add one field to the parent table with a reference to your industry table, minimizing the impact on the standard Microsoft Dynamics NAV table. This can be used in conjunction with developing smaller componentized forms. Although this practice works well with master file–type information, it may not work with journal-type tables, as you may have to change additional code in posting routines.

- **When working with OptionString properties on fields, leave room for Microsoft to add new options in future releases.** Leaving room minimizes the upgrade work you will have to do if there is a clash. Remember that moving your option and update data can be a difficult task.

    Consider this OptionString:

    > Red,Yellow,Brown

    If you add a new option:

    > Red,Yellow,Brown,Pink

    And then Microsoft adds this new string:

    > Red,Yellow,Brown,Blue

    The Microsoft string will override your data.

Instead, you should add to your string witspaces.

> Red,Yellow,Brown,,,,,,,,,,,,,,,,,,,,,,Pink

The amount of spaces you add is up to you and how volatile the field could be, perhaps somewhere in the order of 10 to 20.

- When developing the data model, consider the needs of keys/indexes and the number and user of fields. Refer to the existing documentation, such as the *Performance Troubleshooting Guide* and the SQL Server Resource Kit, for guidance on performance of tables and database design.

Forms

You will have to build a number of forms for your industry solution. The C/SIDE development environment provides an excellent form designer for rapidly developing a user interface for your application. The forms that you develop in the C/SIDE environment are executed in the Microsoft Dynamics NAV Client running on the users' workstations.

Microsoft Dynamics is moving towards implementing RoleTailored user interfaces that allow users to interact with the business application in a more natural way.

When you are developing the forms and working with program management to specify functionality, be sure to consider the roles usage of each form that needs to be developed, and consider developing in smaller, modular forms. For example, Form 42 from the Sales Order entry form is used primarily by the person entering data in the sales orders. However, account managers and financial controllers also use this form for information. Consider building a separate form for the other role types and presenting the information in a summarized view that can be used in other places within the business application. By developing smaller, componentized forms, you can:

- **Reuse the forms in more places within the application.** For example, in the current version of C/SIDE, you might consider developing functionality in subforms for the different roles and bring those together where they are needed on the main forms.

- **Minimize your upgrade work.** By building smaller, componentized forms, you can upgrade the parts independently, taking away the complexity of having to consider the interaction between the information.

The main point you need to consider when developing an industry solution is the maintenance of your forms. If you modify existing forms below the 50,000 number range, the forms you build may have to be upgraded, in service packs, hotfixes, and new product releases. Consider using the following best practices to maintain forms:

- **Copy complex forms to an object in your industry ID range.** That way you have to change only a menu reference. You will have to evaluate this on a case-by-case basis depending on the functionality you are developing. You can use the Source Analysis features in the Microsoft Dynamics NAV Developer's Toolkit to check for references to the form to understand other objects that need to be changed. If the work that is needed to be done to change references outweighs the work minimized for upgrading, then make the changes to the original Form.

- **Hide controls instead of removing them from forms.** This will make it easier to view changes when you look at your old and changed object in a compare tool. However, if the control is in the middle of a group of controls, then the user interface may not look consistent.

- **When working with the PageNames property on forms, leave room for Microsoft to add new names in future releases.** Adding additional spaces will minimize the upgrade work if there is a conflict, as described previously in the OptionString property on fields. For example:

General,Communication,Invoicing,,,,,,,,,,,MyNewPage

You can't create new visual form controls, such as tree controls, in Microsoft Dynamics NAV 4.0. You can only use the existing 16 controls provided for you in the C/SIDE environment. These are excellent for creating a form that allows the user to interact with the business application. In circumstances in which the existing controls are not enough for the interface you need to provide for the role you are developing, you can add new complex user interface features with the use of external controls or applications. You can see this in the existing application, for example, in the Gantt chart functionality added to Microsoft Dynamics NAV 4.0. You can develop such features in standard development tools such as Visual Studio.

The controls can be either COM or .NET based controls. To be executed by the COM interloper layer provided in the C/SIDE development environment, .NET components have to be wrapped as a COM component. This is provided when you reference a COM component as a variable in C/AL code. If you do develop externally to C/SIDE, be sure you consider the increased requirements on installation. Make sure that the external control or component is installed when the user needs to use this functionality.

Developing an external control may have some upgrade benefits, as you would be linking the functionality into the existing C/SIDE environment with C/AL code. This is easy to upgrade and may mean that the control doesn't have to change in a new version. You must also consider controlling the security of the interface as much as possible, as forms developed in C/SIDE can be secured with the use of the application security provided in Microsoft Dynamics NAV. If you develop externally in .NET, then you can make use of authentication provided by the Microsoft Windows operating system.

Developing a control externally also offers other advantages, such as the ability to include more of the business application into other applications such as Microsoft Office. Perhaps the role you are developing for doesn't need to work in the Microsoft Dynamics NAV Client, but instead prepares sales quotes in Microsoft Office Word. By developing externally to the C/SIDE environment, you can create a much better user experience for that particular user.

Reports

Reports are the objects used to print information from the Microsoft Dynamics NAV application. Reports can also be used to process data to mimic a batch process engine. Best practices include:

- **If you need to make complex customizations to an existing report, consider copying the standard report to an object in your solutions object number range.** You will have to evaluate this on a case-by-case basis depending on the functionality you are developing. Use the Source Analysis features in the Microsoft Dynamics NAV Developer's Toolkit to check for references to the report to understand which other objects need to be changed. If the work involved in changing references outweighs the minimization of upgrade issues achieved by copying the report, then make the changes to the original report.

- **If you have external reporting needs, such as reports that need to be executed by users that don't have access to the Microsoft Dynamics NAV Client, consider using Microsoft SQL Server Reporting Services for transactional type reports.** You can make a connection to the Microsoft Dynamics NAV data using an ODBC connection either on the native database or SQL Server.

- **Consider the C/AL business logic code you add to reports. Can this be separated and placed in a Codeunit?** By placing the business logic in a Codeunit, you can simplify the upgrading of the reports. Think of the report as the presentation system. By placing the code in a Codeunit, you make that code available to be called by other mechanisms. Perhaps a report you write in SQL Server Reporting Services could make a call to the same business logic through the Microsoft Dynamics NAV Application Server. Likewise, if you develop an invoice template in Microsoft Office Word to give

29

the users an easier way to manipulate the layout of the invoice in a familiar tool, you could access the same business logic in different report presentation situations as well.

Dataports and XMLports

Dataports and XMLports allow you to import and export data into your industry application. ISVs use Dataports and XML Ports when developing their industry solutions to move data into and out of Microsoft Dynamics NAV. Dataports allow you to work with traditional fixed format and delimited data, and XMLports allow you to work with XML. You can use the ports to:

- **Integrate to other lines of business solutions.** Because you code C/AL code on a Dataport or XMLport, you can develop a standard import function. For example, if you are developing for an industry that uses a specific line of business solution—let's say a billing system running on another platform—you can use Dataports to bring the data into Microsoft Dynamics NAV in a traditional batch manner. You can also use the objects to send data to other applications and use tools such as the Job Scheduler functionality running in a Microsoft Dynamics NAV Client to import and export on a scheduled basis or trigger these with C/AL code.

- **To provide standard template data migration tools.** Perhaps you are recoding a legacy industry solution and using Microsoft Dynamics NAV as the core ERP to that solution. You can develop Dataports and XMLports to migrate the data from your legacy application.

Codeunits

The *C/AL Programming Guide* and *Application Designer's Guide* provide great references for developing C/AL code. Best practices that ISVs use for Codeunits include the following:

- **Avoid the use of code that will stop processes and wait for user input.** Functions like CONFIRM and STRMENU and others that halt the execution of code should be carefully considered or avoided.

- **When developing functions in objects, consider where a function will be used.** If it is only on an object, create it on the object. If you are not going to be calling the function from other objects, consider making it local by changing the Local property where the function is defined. Then it can be seen and called only in that object, which is, effectively, like making a private variable declaration in other programming languages. An added advantage: If you know the function has to be changed, there is only one object to test. If it were global, then you would be able to see other code referencing the function with a tool such as the Microsoft Dynamics NAV Developer's Toolkit.

- **Consider making a Codeunit in your solution number range that contains your business logic code.** When the code needs to be referenced in standard Codeunits, for example, 80 Sales-Post, reference your Codeunit with a variable and pass the relevant variables that you need. Remember to pass as a Ref variable, which allows you to minimize the code that you have to insert into standard code. It also means that you can work on your Codeunit as a black box and change it as needed. In an upgrade, you will only have to merge the reference to the new version. Where there are multiple changes needed in different parts of a posting Codeunit, consider creating different functions in a new Codeunit and then just reference functions in the posting code. However, in these cases, ensure that you document your code effectively to make it easier for a support engineer to read, as the engineer might have to jump around too much and could lose the flow of your code.

- **Register your .NET assembly as a COM object if you have business logic code in your industry solution that has been developed in the .NET environment.** After the assembly is registered as a COM object, it can be called from within C/AL code via COM Interop. Consider your needs for doing this; perhaps access to external

applications that are simpler to develop with Visual Studio and the .NET Framework is sufficient. You can use this approach with the Microsoft Dynamics NAV Application Server.

- **When developing Codeunits, make sure the ability to call the Codeunit is clearly specified** so that other developers know if they pass in the specified parameters they know what the potential result will be. Consider a services-oriented architecture in the development of your code.

- **Make sure you place all text into Text Constants.** That will enable you to translate your
application to a different language.

- **Always develop your C/AL code in U.S. English.** Make sure variables are created, as per naming conventions, with U.S. English names.

Menu System

The menu system is a way for users to find the functionality you've developed. When working with the menu system, consider the following best practices:

- If you are adding a new set of functionality, create a new menu to appear on the list of functional areas, such as Purchase, Warehouse, and so on.

- If your solution is extending an existing standard Microsoft Dynamics NAV module, add new menu items to the existing menu structure.

- For the effective operation of your industry solution, consider changing the menu structure into a structure that is better suited to user roles in your industry industry.

Using Comments

Regardless of your programming skills, you cannot work on a number of products and remember all the reasons for code changes without writing things down. Commenting on your code is a great help for people who have to continue to develop and support your code.

As a best practice, try to avoid too many inline comments as that will clutter the code. Add a reference number, and then add the additional details in the documentation trigger. The reference number should have some meaning relating back to a design specification or a support tracking number from your support system. When you insert code or remove code, add a marker.

```
WITH SalesHeader DO

 BEGIN

 TESTFIELD("Document

 Type");   TESTFIELD("Sell-to

 Customer No.");

 TESTFIELD("Bill-to

 Customer No.");

 TESTFIELD("Posting Date");


  //>> XJZ1.0.1.2

  TESTFIELD("Document Date");

  //<< XJZ1.0.1.2
```

BUILDING INDUSTRY BUSINESS SUCCESS ON MICROSOFT DYNAMICS NAV

```
IF GenJnlCheckLine.DateNotAllowed("Posting Date") THEN
    FIELDERROR("Posting Date",Text045);
```

In the code example above, a new line of code is inserted. You add the reference number XJZ1.01.2 in the documentation trigger and add a small amount of additional comments. The build of the information should exist in your design specifications or support system. Make sure the same information is in both places.

When you change existing code to implement your industry functionality, avoid deleting standard code use comments to block the code out. This will appear in the source code compare tools to make it easier to identify what you have done.

```
WITH SalesHeader DO
 BEGIN
 TESTFIELD("Document
 Type");   TESTFIELD("Sell-to
 Customer No.");
 TESTFIELD("Bill-to
 Customer No.");
 TESTFIELD("Posting Date");

  //>> XJZ1.0.1.3
  //TESTFIELD("Document Date");
  //<< XJZ1.0.1.3

  IF GenJnlCheckLine.DateNotAllowed("Posting Date") THEN
   FIELDERROR("Posting Date",Text045);
```

It's also important that you use the comments features of your source control application. For example, Microsoft Visual SourceSafe® allows you to add comments when you check out and check in your updated object. This enables you to use the reporting functionality in the source control software to get an overview of changes.

Naming Conventions

When developing C/AL code in Microsoft Dynamics NAV, consistency is important for you and the people that need to support and maintain your code. Make sure you develop to the naming standard that has been set up by program management.

Demonstration Data

Demonstration data gives your sales team and channel partners something to use to demonstrate your solution. It can also be used to give consistency and meaning to training materials.

You should have an automated build process to populate your demo data. Typically, you will be building on the standard demo data that is shipped with Microsoft Dynamics NAV, so be aware of the following:

- **Make sure your data is relevant.** Demo data you provide should be relevant to the industry industry you target.

- **Ensure that you produce a good quality demo.** There should be enough data to demonstrate a valid scenario without using real company names. There must be enough demo data available on transactions to make sure you can show the full capabilities of reports. The data should be accurate; when producing a demo for a sales presentation, be sure balances are correct, process flows work correctly, and data relationships are valid.

- **Translate your demo data.** This provides many advantages in the sales process and for training purposes. It also helps customers see your commitment to their market.

Infrastructure Development

Infrastructure development involves developing the foundation pieces of technology. If you develop in the C/SIDE development environment, the foundation is Microsoft Dynamics NAV. However, there may be components that need to be developed externally to the C/SIDE environment to build a specific feature of your industry solution, for example, new tools for companywide ad-hoc reporting.

If you develop a report in the C/SIDE development environment, you have access to a unique foundation technology that makes it simple to develop, such as SIFT, which allows rapid display and filtering of calculated balance. If a user wants to run a report to be reported to the organization over a low-bandwidth connection, and the user doesn't have access to the Microsoft Dynamics NAV Client, then you have to ask someone to generate the report and send it by e-mail to the user. However, if you extend your solution to use SQL Server Reporting Services and integrate this with Microsoft Dynamics NAV, you can provide a way for users to access and execute the report through the SQL Server Reporting Services Web interface.

As a developer, you may have to provide the foundation for this technology to work. This can be developed through the existing ODBC interfaces, or you can provide a foundation by providing predefined SQL Server Analysis Services cubes that users can use for reporting. This should be considered in the infrastructure for the feature.

This could extend into different platform areas, such as integrations to operating system features. The key is that it is a development task, which requires a different set of development knowledge than developing in C/SIDE.

The Test Role

The goal of the test role is to approve the product for release only after all product-quality issues are identified and addressed.

Test Planning

Testing development is a complex process and is, arguably, the most important part of development —because it is the main quality-control part of your applications. When planning to test, take into account the resources you have available. If you don't have enough resources to test the product completely, consider cutting features to free up time for testing.

Scenario Test Cases

You should develop test cases when you are developing your functionality. Test cases define the limits or tolerances for the functionality being tested. Some best practices for test cases include:

- Document the prerequisites needed to run the test, such as sample data, and so on.

- Document the order of the test if your test case is to be run in a sequence.

- Document input variables for functions.

- Document the expected output. The output of journal posting code will result in ledger entries being generated. You can use reports to test the state of the accounts after entries are run. You could use an existing report, for example, a trial balance, or you can develop new reports for these purposes.

Implementation (Manual and Automatic)

You can work with two types of tests. The first is manual: A user takes the test case and sits with the application and drives it manually. The second is an automatic test, that is, a test that runs with the aid of other software.

Developing automated tests can be done with utilities such as screen macro recording software. Such tools can be difficult to use to test Microsoft Dynamics NAV C/SIDE forms. However, they can record key press sequences, which can be used to automate the test process.

Measuring the Test

Measuring and recording test results allows you to track the progress of testing. Record the bank of tests that are run in a database. If a single test doesn't pass, it can indicate that a build should not be shipped. Testers should have access to the work item system to record a defect when the test fails so that developers are notified that an error exists.

You can make use of the Microsoft Dynamics NAV Code Coverage tool in the Microsoft Dynamics NAV Client to test the percentage of code being tested. If the test or combination of tests is not covering 100 percent of the code, you may be releasing untested code, which will create support issues.

C/SIDE GUI Testing

GUI Guidelines provides a guide for your developers and the user experience team members to provide a consistent interface when developing in the C/SIDE development environment. You can use the GUI Test tool provided on the Microsoft Dynamics NAV Tools CD to test whether your application meets the GUI guidelines. This test was developed before version 4.0 of Microsoft Dynamics NAV, but it provides a baseline tool you can use to test for basic user-interface consistency.

Performance Testing

Performance testing has a number of uses. It enables you to test the limits of your application so that you can set boundaries for your customer and channel partners to control their expectations and avoid overselling your industry solution.

Using performance testing as part of your build testing will notify developers of the parts of the application that can be further developed to improve the application.

Microsoft Dynamics NAV provides tools on the Tools CD to help you with performance testing. For more information, see the *Performance and Troubleshooting Guide*, SQL Server Resource Kit, and Application Benchmark Toolkit.

License Files for Testing

Testing your application as it will be deployed at the customer site is difficult because you can't get a license file to test combinations of functionality. To work around this, use the Microsoft Dynamics NAV permissions system combined with your developer's license. The Microsoft Dynamics NAV Tools CD provides a tool that can be used to import and configure permissions. You can use the security system to configure users and assign permissions to restructure the functionality. This is permitted as long as you promptly delete your developer's license when leaving customer premises. Otherwise, you can request a Test/Evaluation License Key by following the guidelines provided. For more information, visit
https://mbs.microsoft.com/partnersource/documentation/howtoarticles/NAVBusinessConcepts.htm.

Microsoft ISV Partners Program Testing

As part of your software marketing, you can use the Microsoft Partner Program as a quality hallmark for your solution. If you have the solution certified under the testing available in the Microsoft Partner Program, you can use the certification in release and in marketing communications.

The User Experience Role

The goal of the user experience role is to enhance user effectiveness. This role ensures that your organization sets and meets standards for the experience users will have with your industry solution.

Internationalization

In your business plan, you have determined the geographic markets you would like to address with your industry solutions. Taking the solution to those markets could require localization.

Localization can be divided into two parts: product functionality and language. The Microsoft Dynamics NAV product is localized for the following geographic markets:

> Australia, Austria, Belgium, Bulgaria, Canada , Croatia, Czech Republic, Denmark, Estonia, Finland, France, Germany, Greece, Hungary, Iceland, India, Indonesia, Ireland, Italy, Latvia, Lithuania, Malaysia, Mexico, Netherlands, New Zealand, Norway, Philippines, Poland, Portugal, Romania, Russia, Serbia and Montenegro, Singapore, Slovakia, Slovenia, Spain, Sweden, Switzerland, Thailand, Turkey, United Kingdom, United States

There is a base Microsoft Dynamics NAV database for the markets listed above. If you want to take your industry solution to a different market, you may have to port some of your development objects to a different localized database.

Does the functionality in your industry solution need to be localized? You are the person best qualified to answer that question, as the answer will vary according to different functionality in different industries. If the market you are going into has more government regulations than the countries you already work in, your solution may require additional localization.

Working with Regional Databases

Localized versions of Microsoft Dynamics NAV are based on a number of regional versions. The regional versions are based on a version developed by core development, the W1 version. If you plan to take your solution into different countries, you must base it on a regional version. This may allow you to develop once and then target a number of countries.

The regional versions are divided as follows:

- Eastern Europe

- Western Europe

- Asia

- North America

If you are targeting your industry solution to a number of countries that would require using different regional databases, you could first develop in the W1 database and then merge your industry functionality into a regional database and perform subsequent testing.

Working with Languages

Markets such as Europe, Asia, and Latin America won't accept a product that is only in English, because the end-user will have difficulty using it. Always develop in U.S. English, and then have that version translated.

Creating a New Language

If your target markets are not covered by a base language already shipped by Microsoft, you will need to request help from Microsoft. The Microsoft Dynamics NAV Client uses two files called FIN.ETX (error texts) and FIN.STX (standard texts). These are sealed in the localization

process by Microsoft and determine the base system texts for the installation. To develop your solution for a specific market, use the base client that contains the language you need. If the language files don't exist, you will need to work with Microsoft to create the files. Otherwise, your solution will be in the language of your targeted market, but the system and error texts will be in another language.

The Microsoft Dynamics NAV Client provides the mechanism to work with languages. For more information, refer to the *Application Designer's Guide*.

If you are working with external controls developed in .NET, you will have to use the globalization features of the .NET environment. If your application has a user interface, you can provide the language as a parameter in initialization so that it can be passed from the calling C/AL code.

Translations

Your industry solution may require that you change the standard definition of an entity in Microsoft Dynamics NAV, for example, changing "Customers" to "Clients." To do this, you will need to use steps to translate the application. Remember to translate online help and training materials so that your customers are aware of the change. If you are instructing people to take standard Microsoft Dynamics NAV courses as prerequisites to using your industry solution, be sure they are aware there is a change in the base application to avoid confusion.

Create a list of terminology (and the context of that terminology) that needs to be changed so that when a new version of Microsoft Dynamics NAV arrives, you can translate the base application for your needs.

Tools

The Microsoft Dynamics NAV Client provides the tools for working with languages. Generally, you will translate the complete application or at least the new objects that you have created and modified for your industry functionality.

To extract the translation texts from Microsoft Dynamics NAV, use the Translate features in the Microsoft Dynamics NAV Client. In the Microsoft Dynamics NAV Client, you can use the Object Designer to filter down to your objects. (On the **Tools** menu, select **Translate**, and then click **Export**.) This will give you the option to export the translation texts to a text file. You can then have this file translated.

The Microsoft Dynamics NAV Localization Workbench (NLW), located on the Microsoft Dynamics NAV Tools CD, allows you to create a project to translate your captions to a new language. The input for the NLW is the Translate Export file created from the Microsoft Dynamics NAV Client. After you have used the NLW to translate the text, you can export to the language you need.

The NLW generates a file you can then use to import back into Microsoft Dynamics NAV. (On the **Tools** menu, select **Translate**, and then click **Import**.) Make sure you select the objects to be updated before executing this menu.

Be careful when you import a translation back into Microsoft Dynamics NAV. All the objects you have selected will have the modified flag, date, and time stamps set to the import time. You can get around this by exporting the object attributes before. For more information see Create a Version List Text File in the Release Management Role section.

License Issues

When developing your localizations, you will need a license file that has access to the different regional databases. You can make a special request to the Microsoft Operations Center to get a different license file to use. To avoid delays, make sure you have a license before you need to start using it.

There are plans to have a global Microsoft Dynamics NAV developer's license, which will make this process simpler.

Technical Communications

Documentation is your means of communicating what your solution does, and how to use and maintain it. Documentation comes in a number of formats, including:

- **User manuals.** User manuals help end users understand the functionality of your solution. They also enable channel partners and end users to train themselves on the functionality. You can find a template for creating user manuals on the Microsoft Dynamics NAV Tools CD.

- **White papers.** White papers describe a particular topic/concept. Marketing white paper would provide a simplified description of a complex concept or technology, whereas technical white papers would provide very detailed examination of a concept or technology.

- **Online help.** Online help gives the end user easy access to descriptions of the functionality they are currently using. The end user knows that F1 is a button they can always press to get help. You can extend and develop the online help for your industry solution by adding an extra help topic. Tools are provided on the Microsoft Dynamics NAV Tools CD, along with an *Online Help Guide*, for creating Microsoft Dynamics NAV online help. Adding online help gives your solution a professional, packaged feel and adds value.

- **Development documentation.** The source control system enables you to control changes to your development. Make sure you use the comments features in the source control program to document why code has changed. This is an important tool because some government regulations, for example, Sarbanes-Oxley, may affect software development, demanding a high level of tracking why code has changed in your development and who has changed it. If you plan to have channel partners customize your industry solution, then provide them with documentation on the entities and process flows in your industry industry. At a minimum, document where you have changed the base Microsoft Dynamics NAV so that adjustments in help, training, and implementation material can be developed.

Training

Training is the way for your solution to be successful, because training teaches people how to use the product and use it in the way you want. Training can come in various forms to help facilitate the delivery of your product. Training forms your readiness backbone, as your product's success depends on the readiness of your end users and partners.

Creating training material is a costly and unique exercise because it typically involves creating training for the base application plus the one-off customizations made at the site. When you are developing a packaged industry product, you have the opportunity to factor the development of the training material into the project. This enables you to recover the costs of developing the material as the material can be used repeatedly.

When developing your training materials, consider your end users and partners:

- End users expect to have training available on new software packages. When you develop an industry solution that is already fulfilling their needs, you can create a regular training schedule. Consider offering training through a training company. That way, you only have to develop the material and then train the trainer at the training company, which frees you from having resources on call to provide end-user training. You should consider online training or e-courses for introductory material. Instructor-led classroom material is the best for learning complex applications like business applications.

- Your channel partners will need training in a number of aspects, including sales training and technical training. Sales training should take the channel partners through the features and functionality of your product so that they can successfully sell it. Technical training staff will go through the product in great detail so that the channel partners' consulting staff know how to implement the solution.

Certification

How extensive are your plans for the rollout of your product? If you have a large product that you plan to support globally and through a partner channel, think carefully about a certification program for your partners.

A certification program has a number of advantages. In the sales process, customers like to have some basic guarantees of quality. If you are selling your product through a channel, the customer wants to know that the product will be installed in a working manner and that they will be supported. They also like to see that the product is supported by the vendor.

With a certification program, you can also classify your partners. This enables you to give different levels of support and to control the delivery of your products.

The Release Management Role

The goal of the release management role is to smooth deployment and ongoing operations.

Infrastructure

The release-management role is important for managing product development. The release manager has control of the systems and processes involved in the production of the solution. One important process is the build process, a formal process that should be run at regular intervals during development. The objective of the build process is to try to assemble all the components of the product to make sure they can work together, and to build a version of the product that can be tested.

Source Code Management, Version Control

An important building block of the build process is an efficient source code management system. Managing objects involves discipline to make sure all the required code is kept safe. When working in the C/SIDE development environment, changes to objects can be recovered after they are saved. If you didn't mean to make a change, you will have to export from an original database and import it into your project database. Employing some basic management practices could save time later in development and prevent losing some of your changes.

What do you put in the source code management tool? You put in everything related to the current product release. That includes product documentation, specification documents, and anything related to developing and shipping the product. That way, you can use the versioning and archiving features to store a complete release at a point in time.

If you have multiple developers working on a project, some code management practices will also help. When working in teams, you can develop on a database stored centrally or develop on a local database. It takes a lot of discipline for team members to avoid overwriting changes that other team members have made. To avoid those issues, you should make use of a source control system.

A source control system includes the following advantages:

- There is a single place where the objects are managed and controlled.

- One developer can work on one object at a time, which avoids issues of losing changes when multiple Microsoft Dynamics NAV developers work on a shared database.

- Object changes that are checked back into the source control can be compared to the base version to see what has changed.

- If incorrect code changes are checked in, then a previous version of the object can be recovered.

- All object changes are logged with the developer who made the change so that you can extract management reports.

- If there is a variation of your ISV project that needs to be created for testing and prototyping, you can branch the source project in the source control tool.

Shared Resources

2.   4.   3.   5.

1.

Microsoft Navision
Client
(FIN.exe or FINSQL.exe)

Base Version

Team Manager

Local Development
Database

Local Check-Out
Folder

7.   6.

Microsoft Navision
Client
(FIN.exe or FINSQL.exe)

Local Developer
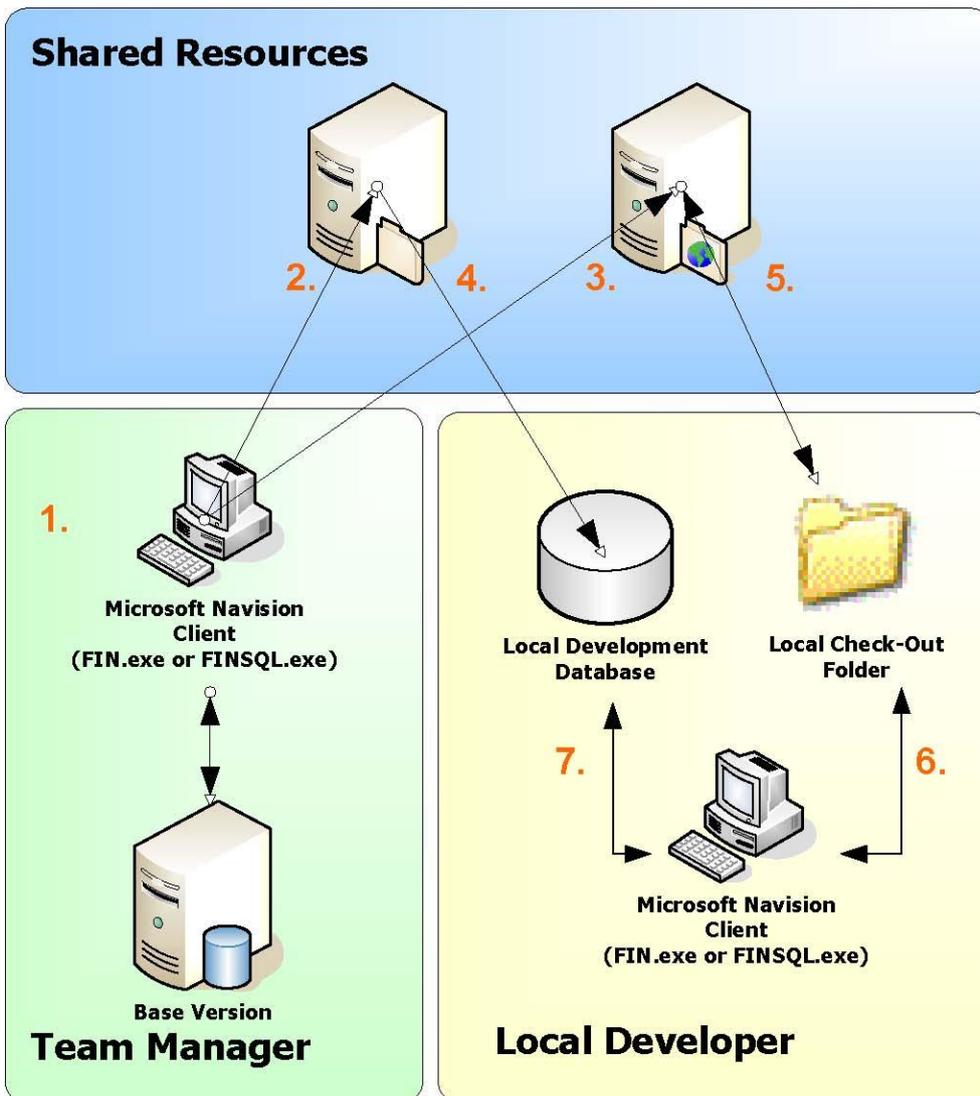
As most Microsoft Dynamics NAV ISV projects involve smaller teams with from 1 to 20 developers, you can make use of Microsoft Visual SourceSafe 6.0. Use the following guidelines for Visual SourceSafe:

1. Create a database with the Microsoft Dynamics NAV base version, which includes any service packs and hotfixes. This will represent your base version for your development project. If hotfixes are released after you build your base version, you will have to merge them into your project.

2. From this database, export all the objects in FOB format. This will enable you to get an original version of an object back if you need it. This action will form a base version with which you can compare. Use this to create changes to documents when you ship the product. Call this file AllObjects.fob. You also check this file into your source control system to use in the daily build process.

3. From this database, export all the objects in TXT format. This will enable you to check the objects into your source control management software. When you export all your objects, this will give you one very large TXT file. To check this into your source control management software, you should work with individual objects. This will take you a long time to import and export from Microsoft Dynamics NAV to set up your source

control project. In the tools folder of the VSS toolkit, you will find the separate utility. This will allow you to split the large TXT file into individual TXT files for each object in the database. You will then check in each of the individual objects. When the developers start working with the product, they will check in and check out individual objects, which will enable the team manager to see who is currently using an object and also to use the functionality of the source control application to see what has changed when developers check in the objects.

4. Build a local development database with a back-up version of the base version.

5. Use the source control application to check in and check out TXT objects to a local check-out folder.

6. Use the Microsoft Dynamics NAV Client to import and export Microsoft Dynamics NAV objects in TXT format from the check-out folder.

7. Developers should use the Microsoft Dynamics NAV Client to develop with C/SIDE on the local database.

Daily Build

You should perform a build daily or, depending on the size of your project, at least two to three times per week. The purpose of the daily build is to make sure that development is on track. If you are not doing a daily build, it is difficult to test, and testing is the most important part of your development cycle.

There are a number of steps in setting up a daily build process. Some may be manual, but you can automate most of the processes to reduce administration, such as starting the process before leaving in the evening. Use the following steps to set up a daily build:

1. **Build an empty database on SQL Server.** You can automate this by using the Microsoft Dynamics NAV Client and build an empty database on SQL Server, for example, called "BUILD". Take a backup of this database and put this aside on the server for use in the daily build. You can restore this using a command line statement with the Microsoft SQL osql utility. Alternatively, you can use the Microsoft Dynamics NAV Client with a macro utility to create a new database either on SQL Server or as an FDB file.

2. **Export all objects from the source control system.** With Visual SourceSafe, you can automate this with a command line with SS.exe. The command line looks like: ss Get $/Navision –R. The project name here is Navision, as it was created in VSS. This will extract the latest version of all the objects from VSS as individual objects. This will also export the AllObjects.FOB file that you create in the setup of VSS.

3. **Create an individual TXT file for each object.** You can make the import process simpler by joining these into one big TXT file. For example, the command line copy *.txt AllObjects.txt when run in the folder with the TXT files extracted from VSS.

4. **Import the AllObjets.fob file.** You won't be able to import the TXT files straight into the empty Microsoft Dynamics NAV database because you don't have insert permissions on your license file to create objects below 50,000. So, using the Microsoft Dynamics NAV Client, import the AllObjets.fob file. Open the Object Designer. On the **File** menu, select **Import all** and locate the AllObjects.fob file.

5. **Use the same import procedure with the Microsoft Dynamics NAV Client to import the AllObjects.txt file.** Importing objects in TXT format will force overwrite of the object, so be careful that you are importing to the correct database.

6. **Compile the objects.** The objects will now all be in an uncompiled state. You can use a macro utility to automate the process. You will have to deal with compile errors manually. The machine on which you run the compile must have access to all the COM objects called by the C/AL code. If you can't set up a machine that includes these,

import the already compiled FOB object into the database. You can create a subset of the FOB objects that you need and import that before you run the compile.

7. **Update your version list.** You can automate this by using a macro that imports an FOB to update the version list.

8. **Update the date and time.**

9. **Build the demo data.** Use the demo data tool that you have created to update the demonstration data. You can automate this with a macro utility that imports and runs an FOB file.

10. **Move the finished database to the test environment.** The build process is finished, and ready for testing.

The remaining sections of this chapter address in greater detail some of the aspects of the build process.

Generating Demonstration Data

Like Microsoft, most ISV partners have a tool that generates demonstration data. This enables the data to be regenerated at any time. You can also translate the demo tool data into another language. Use the following best practices when creating demonstration data:

- Only generate data that populates master file details and the journal tables, and then use posting Codeunits to post the transactions. By posting transactions, you can set the posting data to update the demo data constantly.

- You can make a backup of the standard demonstration data with the back-up features of Microsoft Dynamics NAV. Select the object to not back up objects That way if you use this backup in the build process you will not overwrite your new developments.

Database Consistency

If you are shipping a database to your customers or partners to use in training or demo situations, you want to make sure it is a quality product. Run the database test tools, which you'll find in the Microsoft Dynamics NAV Client (On the **File** menu, select **Database** and then click **Test**). If you have created demo data, be sure that all the tests pass, so that if a customer runs these in a presales demo situation, the test will run without error.

Create a Version List Text File

You can create a version list text file with a Dataport. This is a simple but important tool that can save a lot of time, especially when working with languages. Importing and exporting languages will set the date and time stamps of the objects at the time you import and export. This causes your channel partners some confusion when they look to see what has changed, because a source compare utility will show that all the objects have changed.

Another advantage is that you can maintain your version lists with a spreadsheet and, in this way, add to your documentation.

1. Create a new Dataport.

2. Add the Object table as the DataItem.

3. Add the Dataport fields: Type, ID, Name, Modified, Date, Time, and Version List.

4. Add a filter to the DataItem. Do this by changing the DataItemTableView property. Add the Table Filter as Type=FILTER(<>TableData). This is because the Object table also stores information about the data and you don't want to change this.

5. Run the Dataport as needed to import and export the version list.

A sample object is provided with the toolkit in FOB format. Locate the ExportObjectDetails.fob.

Date and Time on Objects

During the build process, you should update the date and time stamps on your objects for consistency. You should also strip the data and time on objects before checking the TXT object into your source code control tool.

You can update the data and time stamps with C/AL code by using the Object table. A sample piece of code is provided in the toolkit as an FOB object called Create Time and Modified.FOB.

Export Objects in TXT or FOB Format

For more information about exporting objects in TXT and FOB format, refer to the *Application Designer's Guide*.

To export all objects as a TXT format:

1. In the Microsoft Dynamics NAV Client, open the object designer by pressing SHIFT-F12 or on the **Tools** menu, select **Object Designer**.

2. Select **All objects**.

3. In the object designer, press CTRL-A, or on the **Edit** menu, select **Select All**.

4. On the **File** menu, select **Export**. Make sure the Type is Text Format (*.txt).

To export all objects as an FOB format:

1. In the Microsoft Dynamics NAV Client, open the object designer by pressing SHIFT-F12 or on the **Tools** menu, select **Object Designer**.

2. Select **All objects**.

3. In the object designer, press CTRL-A, or on the **Edit** menu, select **Select All**.

4. On the **File** menu, select **Export**. The Type must be Navision Object Format (*.fob).

Local Development or Server Development

Choosing this will depend on how big your development team is and where they are physically located. If your team is all within the same local area network, using a source control system and letting the developer work locally has a number of advantages. They can be mobile, meaning that they can take their work with them and work from a remote location. Another advantage is that multiple developers are not likely to overwrite each other's code, as this is controlled by the source control system. Development with a database on a shared server database is difficult to manage with multiple developers because there is no functionality in C/SIDE to lock an object while it is being developed. This could result in developers overwriting each other's objects as they develop, which results in confusion and downtime recovering the missing code.

Version Numbering

Version numbering is about managing the version of your product so that you can keep track of it. If you know the version, developers and support staff can communicate issues; sales and marketing staff know what features are included in the version they're selling; and partners know what version they need.

Microsoft Dynamics NAV provides a field in the object designer that allows you to update the version list. This is a simple text field that can give your channel partners and support staff a simple way to identify the version of the object deployed. As a guide, version lists should be maintained in the manner shown in the following table.

Release Type	Release Version	Industry Objects

Release	4.00	4.00,ISV1.00
Service Pack	4.00	4.00,ISV1.00.02
Hotfix	4.00	4.00,ISV1.00.02.01

Create a three-letter prefix for your application, and then append this to the objects that you modify in the development process. That way, if you update the object in a service pack update, it appears in the version list with an addition number sequence. If you update the object during a hotfix, add an additional number.

Support

Sustained engineering efforts require that you have a support system to assist ongoing development and communication to channel partners and consultants. Set up a separate support team for your developers. The support team can contain a number of different levels, including:

- Level 1: Provides first-level support to take customer support.

- Level 2: Provides a backup for Level 1 escalations.

- Level 3: Provides a staff of developers who can provide fixes after analyzing support issues. This level provides sustained engineering to allow software defects to be fixed, so that the customer experience is maintained. It also allows product developers to continue working on the upcoming version so that there are no delays.

You can purchase a support system to help you manage support tasks. The following features are a few of the recommended support strategies put in place by other ISVs.

- **Allow customers to enter support requests and software feature requests.** Provide a support number at the initialization of the request and use this number throughout the entire support process.

- **Provide Service Level Agreements (SLAs) to your customers** so that they have documentation of the return cycle of the support request.

- **Attach the support number to comments in code hotfixes.**

- **Use support numbers to communicate changes** that are included in a service pack or hotfix, and document these changes with your product release.

Commercial Release Management

Commercial Release is about getting your product into the market. This phase includes a number of tasks, such as installation information, creating product CDs, and compiling release documentation.

Installation Procedures and Tools

There are a number of important steps to installing and implementing Microsoft Dynamics NAV.

You should encourage channel partners and consultants to take the implementation courses for Microsoft Dynamics NAV. Customers will appreciate a simple and quick product installation because it helps produce a fast ROI. You can simplify the installation process by completing the following steps:

- **Produce a product CD that includes an installation for channel partners.** This CD includes the Microsoft Dynamics NAV Client that will be used to install the solution at the customer site. Most ISVs distribute products to partners in an online download format.

- **Ensure a rapid installation.** Use the Rapid Implementation Methodolody (RIM) tools and best practices provided in the RIM Toolkit for Microsoft Dynamics NAV. Consider customizing the tools for your industry solution to minimize installation time for customers and partners.

- **Minimize end-customer customizations.** Customers like to have a system that is tailored for their business needs. Most of the time they find it difficult to justify upgrading to a new version because of customizations. If you are providing enough industry functionality in your solution, you can minimize end-customer customizations, which helps to simplify the installation process. This also allows you to minimize the requirements for moving the customer to a new version.

- **Provide installation information for customizations**. If you provide custom controls or external applications outside of the Microsoft Dynamics NAV C/SIDE environment, such as an application that runs on the .NET Framework, be sure you provide a program to install this code. In addition, you can use tools like InstallShield to create packaged installers for your solution.

Release Documentation

When you ship your solution, you should also provide documentation. That documentation should contain at least:

- Changes documents, which allow customers and partners to see what has changed in the product release.

- Product release documentation that discusses the functionality in greater detail. This documentation might include fact sheets, data sheets, product documentation, or training materials that describe the solution in greater detail.

Appendix A: For More Information

Microsoft Dynamics NAV—General Resources

Download Microsoft Dynamics NAV from MSDN Subscriptions at
 http://msdn.microsoft.com/en-us/subscriptions/default.aspx or
https://msdn.microsoft.com/subscriptions/.

Download the latest CDs, service packs, and hotfixes from PartnerSource at
http://www.microsoft.com/dynamics/partnersource.mspx

Training
www.microsoft.com/dynamics/using/training.mspx

The Microsoft Dynamics Community
http://www.microsoft.com/dynamics/nav/community.mspx

DemoShield
www.installshield.com/products/demoshield/

Methodologies and Best Practices

Microsoft Solutions Framework (MSF)
http://www.microsoft.com/downloads/details.aspx?familyid=A71AC896-1D28-45A4-880C-8B0CC8265C63&displaylang=en

Sure Step
https://mbs.microsoft.com/partnersource/newsevents/news/newsgeneral/introducingdynamicssurestep.htm?printpage=false

RIM
https://mbs.microsoft.com/partnersource/deployment/methodology/projectmanagement/rapidimplementationmethodology.htm?printpage=false

Business Resources

Microsoft Dynamics ISV Developer Registration Agreement (ISV DRA)
http://www.microsoft.com/downloads/details.aspx?FamilyId=583A26FA-9562-4C42-B8CE-FE64BCF7CB3C&displaylang=en.

Microsoft Solution Provider Agreement (SPA)
https://partner.microsoft.com/global/productssolutions/dynamics/solsprovideragreement.

Add-on Solution Registration
https://mbs.microsoft.com/partnersource/deployment/documentation/howtoarticles/NAV_AddOn Registration.htm

Microsoft Partner Program Resources

Microsoft Partner Program Marketing Resources
https://partner.microsoft.com/global/salesmarketingsection/smcampaigns
https://partner.microsoft.com/global/partner

MBS Competency
https://partner.microsoft.com/global/program/competencies/businesssolutions

ISV/Software Solutions Competency
https://partner.microsoft.com/global/program/competencies/isvsolutions

ISV/Software Solutions Competency Testing
https://partner.microsoft.com/global/program/competencies/isvsolutions/40013116

Other Marketing Resources

Partner Solution Profiler
https://partner.microsoft.com/global/solutionprofiler

Channel Building Resources
https://partner.microsoft.com/global/program/competencies/40045529

Test/Evaluation License Keys
https://mbs.microsoft.com/partnersource/documentation/howtoarticles/NAVBusinessConcepts.htm

Developer Resources

Microsoft Dynamics NAV Developer's Toolkit

Microsoft Dynamics NAV Tools CD, including

- *C/AL Programming Guide*

- *Application Designer's Guide*

- *Terminology Handbook*

- GUI guidelines

- Industry terminology

Installed with the Microsoft Dynamics NAV Client

- *C/SIDE Reference Guide*

Security

General
www.msdn.microsoft.com/security

Writing Secure Code, *(Second Edition), by Michael Howard and David LeBlanc*
www.microsoft.com/MSPress/books/5957.asp

About Microsoft Dynamics

Microsoft Dynamics is a line of financial, customer relationship, and supply chain management solutions that help businesses work more effectively. Delivered through a network of channel partners that provide specialized services, these integrated, adaptable business management solutions work like and with familiar Microsoft software to streamline processes across an entire business.

For information about Microsoft Dynamics go to: www.microsoft.com/dynamics.

Microsoft

BUILDING INDUSTRY BUSINESS SUCCESS ON MICROSOFT DYNAMICS NAV