

**Development Design Document**

|  |  |
| --- | --- |
| ****Title:**** | Printing an Inventory Verification tag from Count Tag Entry program |
| ****Case:**** | 3029 |
| ****Quote:**** | 10587 (8hrs) |
| ****Customer:**** | Minus Forty |
| ****Revision:**** | 1 |
| ****Author:**** | Tom Enns |

# The Six S Development Process

*This process has five steps. You only need to perform one of them now. (You’ll also need to sign off on this document later.)*

### **Request**

Please fill out the **Request** section yourself or in collaboration with a Six S consultant. This is where you will explain your needs along with answering a few supporting questions.

### **Response**

The Six S Strategic Services department (i.e., the development group) will fill out this section. They will summarize your request as they understand it and explain the solution that is being proposed. This document will then be returned to you with an accompanying quote.

**Thoroughly review this section to ensure the development group has accurately understood your issue and that the suggested approach will meet your needs**. This review process is critical to ensuring your development project is successful.

If you have any questions or concerns after your review or any explanations are unclear, please contact Six S BEFORE you approve the design so they can be addressed.

### **Signoff – Design Authorization**

The **Design Authorization** section should be signed upon your approval of the suggested design. This acknowledges that you agree that Six S has understood your request correctly and the design outlined in step 2 (red section) is acceptable. There are also “initial” boxes in this section. Please initial these boxes indicating you agree to said sections before signing off on the yellow section.

### **Development**

When you return the signed document, your request will be placed on the development schedule. A developer will then complete the request and contact you to arrange testing.

### **Signoff – Customer Acceptance**

The **Customer Acceptance** section must be signed upon completion of your development project. This acknowledges that Six S has completed the development work, you have tested it, and you are satisfied with the solution. Development items are not moved to live/production environments until this is signed.

# Request

*Please fill this section out yourself, or in collaboration with a Six S consultant.*

## How is your environment hosted?

*The way your environment is hosted can change how we approach a development task.*

Hosted - Multi-Tenant

## Which database would you like this developed in?

*Six S normally performs new development in the “Test” database. If you prefer another database, please specify. Development is NEVER performed in the Live database. Please ensure your chosen database has current data or let Six S know if you would like us to refresh it.*

Pilot

Please refresh this database with a copy of my live database.

## Which UI do you use?

*If you use multiple UIs, please select the one where you would like this request developed. Note, that selecting “Both” will likely increase the required development effort.*

Kinetic

If you selected other, please explain:

Click or tap here to enter text.

## What is your need?

*Please describe the problem you are solving. Include any related requirements, constraints, complicating factors, or other relevant details. More detail is always better than less.*

***Example:*** *“We have a problem with orders being shipped incomplete but cannot use the standard functionality. We need to prevent partial shipments, but only for specific customers. These customers are identified by a custom checkbox that has already been added to our system.*

*There should be a way for a supervisor to override this logic for exceptional packs.”*

Print an *Inventory Verification* tag from the Count Tag Entry program screen.

Inventory on the floor is counted and recorded (handwritten) on a hard copy of the Count Tag. These hard copy Count Tags are printed from the Count Cycle Maintenance program – (mentioned as a reference to the report style that may be used for this case). These handwritten Count Tags are then keyed into the Count Tag Entry program. It is at this point that there is a need to print an *Inventory Verification* tag, which will be affixed to the item just counted on the floor as proof to any auditor that the item has been counted. As mentioned, the *Inventory Verification* tag report form/style has already been designed by Minus Forty QBD, but we need Six S to develop the ‘print’ button functionality to send it to a printer.

*Detailed steps:*

1. User counts the inventory on the floor.
2. User handwrites the details onto the original Count Tag (hard copy).
3. The original Count Tag is brought back to an MES station where the data is entered into the Count Tag Entry program.
4. After keying in the data, the user has to click a print button to send the ‘*Inventory* Verification’ tag to a printer.
5. The user then takes the *‘Inventory Verification’* tag and affixes it to the counted inventory as proof that it has been counted.

## Do you have technical requirements?

*This section is Optional.   
Six S follows best practices and their best judgement when making technical design decisions. However, if you would like a particular approach utilized (or not) please let us know.*

***Example:*** *“We would like all custom business logic placed within Epicor Functions as we plan to reuse it in other solutions.”*

Explain any technical requirements here.

# Response

*This section will be completed by the Six S development group.* ***Please review thoroughly to ensure your request has been accurately understood******and the proposed solution will meet your needs****.*

## Problem *(what we think you need)*

Initial: \_\_\_\_\_

*This is the Six S summary of your request. We are rewriting what we think you want in our own language.* ***We will develop your solution based on this summary.***

Minus Forty needs a way to trigger the printing of Inventory Verification tags from within Count Tag Entry in Epicor. Right now, they're counting inventory manually, jotting down the details on a physical Count Tag, and then entering this info into the Epicor system. They want a "print" button within the Count Tag Entry screen to print out an Inventory Verification tag as a final step, making their audit process smoother.

## Solution *(how we plan to solve your need)*

Initial: \_\_\_\_\_

*This is the solution Six S is proposing.*

To address this need, we will add a "Print Inventory Verification Tag" button on the Count Tag Entry program screen in Epicor. This button will trigger the printing of an Inventory Verification tag directly to the printer associated with the workstation where the entry is being made. This button will print a tag for the current entry in the count tag entry screen. If no entry is available an error will be thrown stating as such. Clicking the button multiple times will result in multiple tags being printed.

**Assumptions:**

* The tag design already exists, so we just need to hook up the printing functionality.
* The tag is a Bartender tag and we will be printing to a Bartender printer.

## Components

*These are the individual items that makeup the proposed solution.*

|  |  |  |
| --- | --- | --- |
| Name | Type | Purpose |
| Print Button | App Studio Layer | Add a button to the count tag entry screen to allow for tag printing |
| Trigger Print | BPM | Print the Inventory Verification Tag via Bartender |
|  |  |  |

## Unit Tests

Initial: \_\_\_\_\_

*These are the key items that will be tested along with expected results. An acceptable solution will pass all of the tests below.*

|  |  |
| --- | --- |
| Test | Result |
| Enter a count and hit the print inventory tags button | The Tag should print with details of the currently entered part, it should print to the bartender printer that is assigned to the current workstation. |
|  |  |
|  |  |

# Terms

**Design and Engineering Approach**

Epicor’s standard development tools and recommendations along with software development best practices have been utilized wherever possible. This is done to minimize changes to the base software and make upgrades as easy as possible. Anywhere an external or non-standard approach has been suggested, will be clearly noted in the design above.

Every effort has been made to ensure that this solution will exhibit performance in line with base software.

**Upgrade Considerations**

Work required to migrate customizations from one version to another will incur additional costs not covered unless a Customization Maintenance contract has been engaged.

Six S designs solutions with upgradeability in mind. Whenever possible, solutions will be designed to upgrade without the need for manual intervention by a developer. However, Six S can not make guarantees regarding compatibility with future software versions.

**Limitations**

If there is more than a 10% increase to the volumes of data or a change to the typical use of the custom solution as shown by this design, it may render the architecture of the solution inappropriate and/or affect performance.

If you require assistance from Six S Partners in this scenario, any optimization or changes to the custom solution may be chargeable and are not covered by either the original cost of development or Customization Maintenance.

**Testing and Quality Assurance**

Whenever possible, testing is carried out within your own environment. (e.g., Your Test or Pilot database.) When this is not possible, the solution will be tested on the same release version.

Six S Partners ensures that the solution will pass all listed Unit Tests prior to delivery.

**Warranty**

Six S warrants that the product furnished pursuant to the approved Development Design Specification for a period of ninety (90) days from the date of delivery to the customer. The warranties specified in this Section will not apply to defects attributable to: (i) modification of Product without Six S prior written approval, (ii) change of scope or process agreed upon in the documentation to which the product no longer functions, (iii) upgrades, or (iv) due to internal infrastructure viruses and changes. After ninety (90) days the custom maintenance period will be in affect if selected.

# Design Authorization

Any of the functional sections within this design document which are subject to interpretation and may subsequently cause a problem in the software by not satisfying Minus Fortyrequirement, need to be fully documented within this design before approval, otherwise Six S Partners’ development group reserves the right to follow their interpretation.

I have read and initialed (where indicated in yellow) that I agree to the Six S interpretation of our requirements and the Six S proposed solution(s). I have reviewed the entirety of this design document and I agree that it is a fair representation of Minus Fortyrequirements and a solution to them.

|  |  |
| --- | --- |
| **On behalf of Minus Forty:** | |
| **Signature** |  |
| **Date** |  |
| **Print Name** |  |
| **Job Title** |  |

# Customer Acceptance

I have reviewed, tested, and approved the design and function of the solution with changes, additions, deletions, or corrections as agreed upon.

I declare the solution complete and authorize it to be moved to our live environment. (Six S will not deploy any solutions to the live environment prior to explicit scheduling with customer.)

|  |  |
| --- | --- |
| **On behalf of Minus Forty:** | |
| **Signature** |  |
| **Date** |  |
| **Print Name** |  |
| **Job Title** |  |