AIM

MD.070 Application Extension Technical Design

<Company Long Name>

<Subject>

Author: <Author>

Creation Date: April 24, 1999

Last Updated: June 2, 1999

Document Ref: <Document Reference Number>

Version: DRAFT 1A

1. **Title, Subject, Last Updated Date, Reference Number**, **and** **Version** are marked by a Word Bookmark so that they can be easily reproduced in the header and footer of documents. When you change any of these values, be careful not to accidentally delete the bookmark. **You can make bookmarks visible by selecting Tools->Options…View and checking the Bookmarks option in the Show region.**

**Approvals:**

|  |  |
| --- | --- |
| <Approver 1> |  |
| <Approver 2> |  |

1. To add additional approval lines, press [Tab] from the last cell in the table above.

 Copy Number \_\_\_\_\_

1. You can delete any elements of this cover page that you do not need for your document. For example, Copy Number is only required if this is a controlled document and you need to track each copy that you distribute.

## Document Control

**Change Record**

5

| Date | Author | Version | Change Reference |
| --- | --- | --- | --- |
|  |  |  |  |
| 24-Apr-99 | <Author> | Draft 1a | No Previous Document |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

**Reviewers**

| Name | Position |
| --- | --- |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

**Distribution**

| Copy No. | Name | Location |
| --- | --- | --- |
|  |  |  |
|  | Library Master | Project Library |
|  |  | Project Manager |
|  |  |  |
|  |  |  |

1. The copy numbers referenced above should be written into the **Copy Number** space on the cover of each distributed copy. If the document is not controlled, you can delete this table, the Note To Holders, and the **Copy Number** label from the cover page.

**Note To Holders:**

If you receive an electronic copy of this document and print it out, please write your name on the equivalent of the cover page, for document control purposes.

If you receive a hard copy of this document, please write your name on the front cover, for document control purposes.

Contents

Document Control

Technical Overview

Approach

Module List

<Form Title> - Form Logic

Navigation Logic

Block Relationship Diagram

Table and View Usage

Zone and Field Summary

Special Logic

<Report/Program Name> - Concurrent Program Logic

Calling Arguments

Log Output

Table and View Usage

Program Logic (pseudo code)

SQL Statements

Default Data Sources

Validation Logic

Incompatibility

Performance Considerations

Other Considerations

Integration Issues

Changes Required

Shared Components

Alert conditions

Incompatibilities

Performance Issues

Database Design

Desired Table Changes

New/Updated Seed Data

Descriptive Flexfields

Value Sets

Grants/Synonyms

Archiving

Database Diagram

Tables, Indexes, Sequences

Installation Requirements

Implementation Notes

Design Summary

Coding Summary

Testing Summary

Installation

Open and Closed Issues for this Deliverable

Open Issues

Closed Issues

1. To update the table of contents, put the cursor anywhere in the table and press [F9]. To change the number of levels displayed, select the menu option Insert‑>Index and Tables, make sure the Table of Contents tab is active, and change the Number of Levels to a new value.

## Technical Overview

This document defines the technical components required to implement customization <BRM Number>, <Subject>. This Application Extension Technical Design document complements the Application Extension Functional Design document for <Subject> and you should consider the set to be the complete detailed design.

### Approach

1. If the Functional Design included a Technical Overview section, duplicate that information here and add additional details.

### Module List

#### Forms

<Subject> includes the following forms:

**<Form Name>** <Description>

#### Reports

<Subject> includes the following reports:

**<Report Name>** <Description>

#### Concurrent Programs

<Subject> includes the following concurrent programs:

**<Program Name>** <Description>

#### Database Triggers

<Subject> includes the following database triggers:

**<Trigger Name>** <Description>

1. Add other components above as required

## <Form Title> - Form Logic

<Overview description of form logic>

### Navigation Logic

1. Update the sample navigation logic below as needed to reflect the requirements of your customization.

#### Entry Mode

**START**

1. User enters data into form.
2. User commits transaction.
3. Form calls concurrent program X with selected parameters.
4. Form displays "Request xxxx has been sent to the concurrent manager".
5. User presses any key to acknowledge message.
6. Form clears.
7. Cursor moves to field 1.

**END**

### Block Relationship Diagram

The diagram below represents the base tables of each block or zone of the form (vertical) and tables referenced for validation or lookups (horizontal).



1. The diagram above is just an example—Double click on it to edit it as an embedded object. Select it and choose Edit->VISIO Object->Open to open it in the full Visio application.  
     
   You can also use Oracle Developer to draw your diagram and define most of the information in this section.

### Table and View Usage

| Table Name | Select | Insert | Update | Delete | Base Table |
| --- | --- | --- | --- | --- | --- |
|  | X |  |  |  |  |
| MTL\_SYSTEM\_ITEMS |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |

### Zone and Field Summary

#### Zone 1

Base Table:

Where/Order By:

| Field/Column | Type | Rqd? | Default | Validation | QuickPick |
| --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |
| Field 1  COLUMN\_NAME | Char | Yes |  | Validated against <column>  from <table>  where <column> = <value> | Display MEANING  from <Application Short Name>\_LOOKUPS where LOOKUP\_TYPE = '<lookup type> |
|  |  |  |  |  |  |
|  |  |  |  |  |  |

#### Zone 2

Base Table:

Where/Order By:

| Field | Type | Rqd? | Default | Validation | QuickPick |
| --- | --- | --- | --- | --- | --- |
|  |  |  |  | ( |  |
| Field 1  COLUMN\_NAME | Char | No |  | (QuickPick validation) | Display MEANING  from <Application Short Name>\_LOOKUPS where LOOKUP\_TYPE = '<lookup type> |
|  |  |  |  |  |  |

### Special Logic

#### Zone 1

<Description of special zone-level logic (if any)>

##### Field Name

<Description of special field logic>

## <Report/Program Name> - Concurrent Program Logic

<Overview description>

### Calling Arguments

<PROGNAME> is called from the Standard Report Submission form with the following arguments:

| Argument | Prompt | Value Set | Default Value |
| --- | --- | --- | --- |
|  |  |  |  |
| ORG\_ID | Organization id | Generic | :$PROFILE$.mfg\_organization\_id |
| USER\_ID | Application User ID | Generic | :$PROFILE$.user\_id |
| DATE | Selection Date | FND\_Date4\_required | :$$DATE$$ |

Example of call from a form:

#FND CONCURRENT <Application Short Name> <PROGNAME>

":GET\_ORGANIZATION.ORGANIZATION\_ID"

":LAST\_UPDATED\_BY"

Example of call from Unix:

<PROGNAME> mfg/mfg 1 0

### Log Output

.........1.........2.........3.........4.........5.........6.........7.........8

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

### Table and View Usage

| Table Name | Select | Insert | Update | Delete |
| --- | --- | --- | --- | --- |
|  |  |  |  |  |
| MTL\_SYSTEM\_ITEMS | X |  |  |  |
|  |  |  |  |  |

### Program Logic (pseudo code)

BEGIN

Get command line parameters

Write Report Header

Select rows (see SQL statement 1)

FOR EACH row returned

Build output record(s) (see data sources)

Write row(s) to <table>

ENDFOR

END

### SQL Statements

#### 1 - Data Selection

SELECT <data>

FROM <tables>

WHERE <select criteria>

AND <join conditions>

### Default Data Sources

|  |  |  |
| --- | --- | --- |
| **Table** | **Column** | **Source Value** |
|  |  |  |
| <table name> | <column name | <table>.<column> |
|  | <column 2> | 'Literal string' |
|  | <column 3> | (Description of derived value) |

### Validation Logic

#### Error Conditions:

#### Warning Conditions:

### Incompatibility

### Performance Considerations

### Other Considerations

#### Restart Strategy

#### Crash Recovery

## Integration Issues

### Changes Required

#### Within Product

Custom menus

#### Other Products

No changes required

### Shared Components

1. The Oracle Application product short names listed below are examples only. Update the shared component information to reflect the requirements of your customization.

#### INV

None

#### BOM

None

#### ENG

None

#### MPS/MRP

None

#### WIP

None

### Alert conditions

None.

### Incompatibilities

None.

### Performance Issues

None.

## Database Design

This section summarized new and changed database objects and data required to support <Subject>. However, the complete database design is documented in the Database Extensions Design document.

### Desired Table Changes

None

### New/Updated Seed Data

Rows added to <App Prefix>\_LOOKUPS:

|  |  |  |
| --- | --- | --- |
| Lookup Type | **Code** | **Meaning** |
|  |  |  |
|  |  |  |

### Descriptive Flexfields

Application:  
Form Name:  
Base Table:

| Segment Name | Prompt | Size | Value Set | Default Value |
| --- | --- | --- | --- | --- |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

Application:  
Form Name:  
Base Table:

| Segment Name | Prompt | Size | Value Set | Default Value |
| --- | --- | --- | --- | --- |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

### Value Sets

| Value Set | Size | Type | Rqd | Validation/Values |
| --- | --- | --- | --- | --- |
|  |  |  |  |  |
| <App Prefix>\_ITEM\_SELECT | 15 | Char | Yes | ALL All Items ONE Specific Item |
| <App Prefix>\_SPECIAL\_CODE | 10 | Char | No | Code Meaning  From <App Prefix>­\_LOOKUPS  Where lookup\_type = 'SPECIAL' |

### Grants/Synonyms

|  |  |  |
| --- | --- | --- |
| Owner | Object | Grantees |
|  |  |  |
| <App Prefix> | <App Prefix>\_LOOKUPS | MFG INV |
|  |  |  |

### Archiving

No need to archive.

### Database Diagram

No new tables

### Tables, Indexes, Sequences

No new tables

## Installation Requirements

Installation scripts must be prepared to perform the following actions in an automated way:

1. Create new tables.
2. Insert seed data into <App Prefix>\_LOOKUPS as described above.
3. Run grant/synonym script.
4. Define Value Sets and Validation Tables.
5. Define Descriptive Flexfields.
6. Define Help text.
7. Define Message text.
8. Register Forms.
9. Register Concurrent Programs.
10. Register Standard Report Submission parameters.
11. Create Menus.
12. Add to or modify this list as appropriate. Provide additional details where necessary to facilitate the creation of the installation routines.

## Implementation Notes

Complete this section after completing the <Subject> customization. Update the design to reflect any changes made in the course of development and describe how the customization was implemented below.

This document describes exactly how <Subject> was developed and implemented at <Company Long Name>.

### Design Summary

#### Functional Design

#### Functional Design Approval

#### Technical Design

#### Design Review

#### Final Acceptance

### Coding Summary

Development was done on <description of hardware>, to work with Release <Applications Release> of Oracle Applications. All coding follows the standards defined in the Build Standards document for <Project Name>.

#### Program Files

The files required for this customization are as follows:

| File | Description | Coded By |
| --- | --- | --- |
|  |  |  |
|  |  |  |

### Testing Summary

The customizations were tested in <Company Short Name>'s test environment before being moved to production. See the Link Test scripts for more information.

### Installation

All modules are installed as if part of a separate product. A UNIX directory structure is created under $APPL\_TOP as shown in the following example:

$APPL\_TOP

|

<Application Short Name>

|

1.0.0

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_|\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

| | | | | | | | |

forms bin src sql log out srw install sql

The directories contain the components of the enhancement as shown below (directories not listed are empty). All future customizations will also be stored in these directories.

forms/

<Formname>

bin/

src/

sql/

srw/

install/sql/

Update the example directory structure above as appropriate for your project. Fill in the filenames under the appropriate directories.

The forms and concurrent programs are registered in Application Object Library under the <Application Short Name> application.

A new set of menus (owned by <Application Short Name>) is created to call the new forms as follows:

Navigate  
 Sub-menu  
 Sub-menu

Indicate the specific menu options added to access new forms.

## Open and Closed Issues for this Deliverable

Add open issues that you identify while writing or reviewing this document to the open issues section. As you resolve issues, move them to the closed issues section and keep the issue ID the same. Include an explanation of the resolution.  
  
When this deliverable is complete, any open issues should be transferred to the project- or process-level Risk and Issue Log (PJM.CR.040) and managed using a project level Risk and Issue Form (PJM.CR.040). In addition, the open items should remain in the open issues section of this deliverable, but flagged in the resolution column as being transferred.

### Open Issues

| ID | Issue | Resolution | Responsibility | Target Date | Impact Date |
| --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |

### Closed Issues

| ID | Issue | Resolution | Responsibility | Target Date | Impact Date |
| --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |