2013

Los Angeles Community College District

[SAP DEVELOPMENT AND QUALITY ASSURANCE] POLICIES AND PROCEDURES
# REVISION HISTORY

<table>
<thead>
<tr>
<th>Date</th>
<th>Updated by</th>
<th>Reviewed by</th>
<th>Document Version</th>
</tr>
</thead>
<tbody>
<tr>
<td>2/27/2009</td>
<td>Imran Asif</td>
<td>Andy Duran</td>
<td>1.0</td>
</tr>
<tr>
<td>4/13/2009</td>
<td>Imran Asif</td>
<td>Andy Duran</td>
<td>1.1</td>
</tr>
<tr>
<td>5/27/2009</td>
<td>Imran Asif</td>
<td>Andy Duran</td>
<td>1.2</td>
</tr>
<tr>
<td>7/2/2010</td>
<td>Imran Asif</td>
<td>Andy Duran</td>
<td>1.4</td>
</tr>
<tr>
<td>01/10/2013</td>
<td>Samin Atiff</td>
<td><strong>Andy Duran</strong></td>
<td>2.0</td>
</tr>
</tbody>
</table>
# TABLE OF CONTENTS

Revision History ........................................................................................................... 2
1. Introduction ............................................................................................................... 8
   1.1 Purpose .............................................................................................................. 8
   1.2 Objective ......................................................................................................... 8
   1.3 Definitions ....................................................................................................... 8
2. Overview .................................................................................................................. 9
   2.1 SAP Development Life Cycle ......................................................................... 9
3. LACCD SAP Landscape .......................................................................................... 12
   3.1 R/3 Landscape .................................................................................................. 12
   3.2 Portal Landscape ................................................................................................ 15
      3.2.1 Portal Landscape ....................................................................................... 15
      3.2.2 External Portal ............................................................................................ 15
      3.2.3 Production Portal ....................................................................................... 16
      3.2.4 Quality Portal ............................................................................................. 16
      3.2.5 Development Portal ................................................................................... 17
      3.2.6 Sandbox Portal ........................................................................................... 17
   3.3 B/W Landscape .................................................................................................. 18
4. ABAP Development .................................................................................................. 19
   4.1. Development Tools ......................................................................................... 19
   4.2 Pre-coding Phase ............................................................................................... 20
   4.3 Coding Techniques ............................................................................................. 21
      4.3.1 Naming Standards ....................................................................................... 21
      4.3.2 Documentation Standards ................................................................. 33
      4.3.3 Formatting .................................................................................................. 35
   4.4 Programming Practices ....................................................................................... 36
1. Introduction

5. Performance Standards
   5.1 General Performance Standards
   5.2 Performance Checking

6. Dictionary: Table Development Standards
   6.1 Table naming convention
   6.2 USER Table Definition Convention
   6.3 Maintenance Settings
   6.4 Delivery Class
   6.5 Access Control - Authorization Group
   6.6 Direct database updates of SAP standard tables
   6.7 Approval Process for Direct Update to SAP table

7. Portal Development Standards
   7.1 Overview
   7.2 Portal Content Objects
   7.3 Portal Objects Naming Standards
      7.3.1 ID Prefix format for Portal Content Objects
   7.4 Transports
   7.5 Jco Destinations
   7.6 Webdynpro for Java Naming Standards
      7.6.1 Development Entities
      7.6.2 Context Entities
   7.7 Portal User Interface Design Guidelines

8. Business Warehouse
   8.1 BW Objects Naming Standards
   8.2 Broken Reports

9. LACCD Standard Email Template

10. Transports Standards
10.1 Overview of LACCD Transport Process................................................................. 104
10.2. Transport Schedule ............................................................................................... 107
10.3 Transports Description Format .............................................................................. 109
10.4 Transport Priority Descriptions ............................................................................ 109
10.5. Transport Request Form ...................................................................................... 110
11. Authorization Standards .......................................................................................... 112
11.1 Authorizations Concept.......................................................................................... 112
11.2 Authorizations: Usage of SAP objects .................................................................. 113
11.3 Restricting Access to HR Data .............................................................................. 113
12. Testing Standards .................................................................................................... 115
  12.1. Testing Phases ...................................................................................................... 117
    12.1.1 Unit Testing ...................................................................................................... 117
    12.1.2 Scenario Testing .............................................................................................. 126
    12.1.3. Integration Testing ........................................................................................ 126
    12.1.4. Performance Testing ..................................................................................... 127
    12.1.5. User Acceptance Testing .............................................................................. 127
    12.1.6. Regression Testing ........................................................................................ 127
13. Issue Management Standards ................................................................................... 128
  Logging on HP Quality Center ................................................................................... 129
  13.2 New Issue Creation ............................................................................................... 129
  13.3 Issue Details Dialog Box....................................................................................... 138
  13.4 Email Notifications .............................................................................................. 143
  13.5 Viewing Issues From the Email .......................................................................... 144
  13.6 Search For an Issue by Issue Title ...................................................................... 145
  13.7 Issue Transition .................................................................................................... 146
14. Requirements Management ....................................................................................... 151
  14.1 Add a requirement ............................................................................................... 152
  14.2 Requirements Traceability Matrix ........................................................................ 155
14. 3 Risk Based Quality Management (RBQM) ........................................................................... 159
15. Documentation standards ........................................................................................................ 161
   15.1 Naming Convention for Documents .................................................................................. 162
   15.2 Shared Drive .................................................................................................................... 163
   15.3 Folder Structure in Shared Drive ...................................................................................... 165
16. Scheduling Background Jobs ................................................................................................ 166
   16.1 Introduction ....................................................................................................................... 166
   16.2 Procedure To Schedule A Background Job ....................................................................... 166
   16.3 Scheduling Jobs With Job Documentation ........................................................................ 167
   16.4 Background Job Scheduling Standards ............................................................................. 167
   16.5 Production Control Considerations ................................................................................. 168
   16.6 Communication Between Foreground and Background Processes .............................. 168
17. ABAP/4 Tuning checklist ....................................................................................................... 171
Appendix A: Sample Programs ................................................................................................. 174
Appendix B: Two/Three-Character Codes for Application Areas ........................................... 207
Appendix C: Development Classes .......................................................................................... 209
Appendix D: LACCD Report Specification Form .................................................................... 211
Appendix E: Project Plan Template ......................................................................................... 217
Appendix F: Technical Specifications Template ....................................................................... 218
1. INTRODUCTION

1.1 PURPOSE

The purpose of this document is to define and maintain a single source of accurate information about LACCD’s SAP Development and Quality Assurance policies and procedures. It is strongly suggested that all staff follow this document closely.

1.2 OBJECTIVE

The objective is to lay out programming guidelines, development and testing standards that should be incorporated by all the team members in the product development process.

1.3 DEFINITIONS

**Quality Assurance (QA)** - As per IEEE, Quality Assurance (QA) can be defined as “A planned and systematic pattern of all actions necessary to provide adequate confidence that the item or product conforms to established technical requirements”1.

QA is the responsibility of the **entire team**, and it involves preventing errors and corrects errors when found.

**Quality Control (QC)** - QC is a set of activities focused on finding defects in specific product or deliverables. It involves finding defects and resolving it. It is the responsibility of the **team member**.

---


1. Introduction

LACCD| Confidential Version 2.0 - 2013
2. OVERVIEW

2.1 SAP DEVELOPMENT LIFE CYCLE

The implementation of a SAP project involves the following stages of ASAP methodology

**Project Preparation:** The first step in ASAP is Project Preparation. In this step, the functional team member analyzes the business process and defines the system for SAP and gives the timeline for the project. The resources required and the budgets are also considered in this step.

It is the responsibility of the developer to conduct technical analysis of the business specifications. Usually at the end of this process, the business specifications are updated and finalized by the Coordinator and technical specifications are prepared by the Developer. Technical specifications should display the input and output of the process and the program logic. Click on the link to view the Technical Specifications document.

If technical specifications are cross-module, they must be signed off by the LACCD technical expert/Coordinator from the appropriate module. At the end of this stage the developer should finalize the technical design of the program and review it with the coordinator.

**Blue Print Preparation:** This is the second step in ASAP Methodology. The Project Preparation scenarios should be finalized and an outline of the Project is the final outcome of this phase.
Realization: Actual development takes place during this phase. The ABAP development team is responsible for this step.

Final Preparation: Testing of the projects is conducted during this phase by the ABAP developer. This is the area where the Unit Test Document is prepared (i.e. Positive Testing and Negative Testing.

GO Live and Support:

This is the final stage where the project is running in the live environment. In this step the bugs if any that are found in the Go Live procedures are fixed and the project will be considered as support project.
Shown below is a typical Application Project Life cycle.

Application Project Life Cycle

1. Project Request from Business
2. Manager approves/disapproves project after review
3. IT Manager notifies the Business
   - More work needed
4. IT and Business user develop and agree on requirements, business signs off on requirements
5. Developer/project team will create design document and project schedule
6. Developer/Project Team develops code, configures system, creates interfaces for the project. Work takes place in the Development/Testing environment
7. Support Team finalizes documentation, and rollout plan and conduct training as needed. Developer works with Basis team on authorizations
8. Project is ready for deployment
9. Project is made available to the business for testing
10. Project ready for evaluation by business
11. Project is tested internally within IT
12. Users are notified of new functionality.
13. Project is deployed to production system

Not Approved

Approved

More work needed
3. LACCD SAP LANDSCAPE

3.1 R/3 LANDSCAPE

LACCD’s SAP R/3 landscape is a three-system landscape which consists of a development system DEV, a quality assurance system QAS, and a production system PRD. Whenever new projects are implemented sandbox systems are provided to the development team and they serve as project specific development servers.

These names and numbers are described by LACCD and are independent of any other names of the servers used elsewhere.

All transport will be moved only after successful testing from DEV to QAS and finally into the production system.
The details of the systems at LACCD are given below:

<table>
<thead>
<tr>
<th>SID</th>
<th>O.S.</th>
<th>DB</th>
<th>Hostname</th>
<th>Java Instance</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>PRD</td>
<td>HPUX</td>
<td>Oracle</td>
<td>doux008</td>
<td>xx</td>
<td>production ERP</td>
</tr>
<tr>
<td>QAS</td>
<td>HPUX</td>
<td>Oracle</td>
<td>doux013</td>
<td>xx</td>
<td>Quality Assurance ERP</td>
</tr>
<tr>
<td>DEV</td>
<td>HPUX</td>
<td>Oracle</td>
<td>doux012</td>
<td>xx</td>
<td>Development ERP</td>
</tr>
<tr>
<td>DV2</td>
<td>HPUX</td>
<td>Oracle</td>
<td>doux1010</td>
<td>xx</td>
<td>Sandbox</td>
</tr>
<tr>
<td>DVL</td>
<td>HPUX</td>
<td>Oracle</td>
<td>doux1027</td>
<td>xx</td>
<td>Future development ERP</td>
</tr>
<tr>
<td>PR2</td>
<td>HPUX</td>
<td>Oracle</td>
<td>doux1004</td>
<td>xx</td>
<td>Sandbox</td>
</tr>
<tr>
<td>SSC</td>
<td>HPUX</td>
<td>Oracle</td>
<td>doux1031</td>
<td>xx</td>
<td>Sandbox ERP 6.0 Ehp6 upgraded</td>
</tr>
<tr>
<td>TAM</td>
<td>HPUX</td>
<td>Oracle</td>
<td>doux1017</td>
<td>xx</td>
<td>Asset Mgmt</td>
</tr>
<tr>
<td>XAM</td>
<td>HPUX</td>
<td>Oracle</td>
<td>doux1030</td>
<td>xx</td>
<td>Asset Mgmt</td>
</tr>
<tr>
<td>ZLD</td>
<td>HPUX</td>
<td>Oracle</td>
<td>doux1035</td>
<td>xx</td>
<td>ERP 6.0 Ehp6 new installation</td>
</tr>
<tr>
<td>PBW</td>
<td>HPUX</td>
<td>Oracle</td>
<td>doux007</td>
<td>&lt;01&gt;</td>
<td>Business Intelligence (production)</td>
</tr>
<tr>
<td>QBW</td>
<td>HPUX</td>
<td>Oracle</td>
<td>doux1005</td>
<td>&lt;01&gt;</td>
<td>Business Intelligence (quality assurance)</td>
</tr>
<tr>
<td>DBW</td>
<td>HPUX</td>
<td>Oracle</td>
<td>doux006</td>
<td>&lt;01&gt;</td>
<td>Business Intelligence (development)</td>
</tr>
<tr>
<td>MBW</td>
<td>HPUX</td>
<td>Oracle</td>
<td>doux1006</td>
<td>&lt;01&gt;</td>
<td>Business Intelligence (sandbox)</td>
</tr>
<tr>
<td>SPM</td>
<td>HPUX</td>
<td>Oracle</td>
<td>zen</td>
<td>&lt;01&gt;</td>
<td>Solution Mgr 7.01</td>
</tr>
<tr>
<td>ZSM</td>
<td>HPUX</td>
<td>Oracle</td>
<td>doux1032</td>
<td>&lt;00&gt;</td>
<td>Solution Mgr 7.1</td>
</tr>
<tr>
<td>QSM</td>
<td>HPUX</td>
<td>Oracle</td>
<td>doux1026</td>
<td>&lt;00&gt;</td>
<td>Solution Mgr 7.1</td>
</tr>
</tbody>
</table>
To summarize, the different systems in LACCD R/3 Landscape are as given below:
Transport requests will flow from DEV ----> QAS ----> PRD and not backwards.

Sandbox ----> Development ----> Quality ----> Production

**Development server** - is where the Developers/Configurators do the customization as per the Company’s requirement. In the initial stages of implementing a project all the development/configuration/customization as per the Company's business process is done in the Development server. LACCD's SAP development server is DEV-300. After the development/configuration has been done, the developers should conduct unit testing and initiate a transport request with the Basis team to move the code from Development server to Quality server.

**Quality Server** - LACCD's SAP testing server is QAS-300. Thorough testing is performed in the Quality Server and saved in workbench requests, to be transported to Production server.

**Production Server:** This is the last/most refined client where the user will work after project GO LIVE. Any changes/new development is done in development client and the request is transported to production.

**Sandbox Server:** LACCD’s sandbox is PR2-300. This client is a copy of Production Server. All new developments should be done in this server; changes to development/configuration are carried out in this server. All transports going to Production will also go into Sandbox server. Sandbox server contains limited data to work on. Only minute changes to development/configuration that will not require UAT (User Acceptance Testing) must be performed in this server (before DEV)

**Project Specific Servers:** There are a number of project specific servers that are assigned at the time of Support Pack Upgrades or new system implementation. These are provided to develop the system in a clean environment and unit tested before moving to DEV.
3.2 PORTAL LANDSCAPE

3.2.1 PORTAL LANDSCAPE

3.2.2 EXTERNAL PORTAL

Windows 2008 R2 (x64) 5.2, RAM- 16GB
Database: Ms/Sql serer 2008 R2 (Intel x64)
J2EE Engine: NW 7.3 EHP1
Portal: 7.3 EHP1
Db<SID>: EME
DB Host: downn1028.dsc
SAP<SID>: EME
SAP Host: downn1028.dsc
SAP Instance: JC00
ITS Host: prd-its.dsc
3. LACCD SAP Landscape

3.2.3 PRODUCTION PORTAL

Windows 2008 R2 (x64), RAM- 16GB (portal)
Database: Ms/Sql server 2008 R2 (Intel x64)
J2EE Engine: NW 7.3 EHP1
Portal: 7.3 EHP1
Db<SID>: PEP
DB Host: prd-db
SAP<SID>: PEP
SAP Host: prd-portal1
SAP Instance: JC00
ITS Host: prd-its.dsc

3.2.4 QUALITY PORTAL

Windows 2008 R2 (x64), RAM- 12GB
Database: Ms/Sql server 2008 R2 (Intel x64)
J2EE Engine: NW 7.3 EHP1
Portal: 7.3 EHP1
Db<SID>: QEP
DB Host: qas-db
SAP<SID>: QEP
SAP Host: qas-portal1
SAP Instance: JC00
ITS Host: dev-qas-its
3. LACCD SAP Landscape

**3.2.5 DEVELOPMENT PORTAL**

![Diagram of Development Portal]

- **Windows 2008 R2 (x64)**, **RAM- 12GB**
- **Database**: Ms/Sql server 2008 R2 (Intel x64)
- **J2EE Engine**: NW 7.3 EHP1
- **Portal**: 7.3 EHP1
- **Db<SID>**: DEP
- **DB Host**: dev-portal
- **SAP<SID>**: DEP
- **SAP Host**: dev-portal
- **SAP Instance**: JC00
- **ITS Host**: dev-qas-its

**3.2.6 SANDBOX PORTAL**

![Diagram of Sandbox Portal]

- **Windows 2003 (x86) 5.2 RAM- 2GB**
- **Database**: Ms/Sql server 2008 R2 (Intel x64)
- **J2EE Engine**: NW 7.3 EHP1
- **Portal**: 7.3 EHP1
- **Db<SID>**: SEP
- **DB Host**: spsbox
- **SAP<SID>**: SEP
- **SAP Host**: spsbox
- **SAP Instance**: JC00
- **ITS Host**: dev-qas-its
The diagram of the LACCD B/W landscape and the transport route is given below:
4. ABAP DEVELOPMENT

4.1. DEVELOPMENT TOOLS

LACCD’s SAP system has the following development tools available:

- **ABAP Workbench** - to develop reports, programs, functions, tables, screens, menus
- **ABAP4 Query** - to develop reports
- **Report Writer** - to develop reports
- **BAPI Development** - to develop BAPI’s for Web and RFC
- **SAPscript** - to develop printed forms
- **SAP Smart Forms** - recommended to develop printed forms

Listed below are the tools used from EPI-Use. Click on the links below for user guides.

- **Data Sync Manager (DSM)** – [User guide](#)
- **Payroll Recon** – [User guide](#)
- **Variance Monitor** – [User guide](#)
4.2 PRE-CODING PHASE

Before starting to code as part of technical review, the developer should:

- Identify similar programs, functions etc., by reviewing the existing functions or reports.
- Identify tables to be used as a source of the information by checking with the Coordinators, review the SAP Logical Databases, Repository Browser, Debug or SQL-trace SAP programs, or Data Architecture documents.
- Decide if new tables or indexes must be created.
- Decide whether to copy an existing program or create a brand-new one.
- Define names for new programs, tables, etc.
- In case of update or conversion programs, determine what the best update method is (call transaction or batch input sessions or inserts). Volume should be an important factor in this decision.
- Decide what development class to use.
- If this is a one-time-only program then saving it in the conversion development class (ZCONVERS) should be considered.

When new objects are created, they should follow naming conventions as described in "Naming Standards" section. If there is no convention for the new object, the developer must contact the SAP ERP Manager.

When a developer applies a SAP OSS note(s), it must be documented when resolved.

The sections below will highlight some of the Coding Techniques and Programming Practices that should be followed at LACCD. The coding techniques primarily improve the readability and maintainability of code, whereas programming practices are mostly performance enhancements.
4.3 CODING TECHNIQUES

Coding Techniques can be divided into the following sections:

1. Naming Standards
2. Documentations
3. Formatting

4.3.1 NAMING STANDARDS

Suitable naming for repository objects defined outside the program and for entities declared within the program is of paramount importance for tracing and comprehending a program. Given below are some of the recommended naming standards to be followed during development process at LACCD.

Primarily, all names should be legible, catchy and suitable. It is also necessary to avoid naming conflicts.

4.3.1.1 ABAP WORKBENCH OBJECTS

ABAP/4 variable names can be up to 30 characters for DATA fields and subroutines, and up to 8 characters for SELECT-OPTIONS and PARAMETERS, therefore, as a standard, make the names descriptive. Do NOT use a hyphen “-“ in variable names or in paragraph (“form”) names. Use the underscore “_“ instead. SAP uses the hyphen “-“ to separate a table name from the name of a field in that table. Using an underscore will increase readability of your program.
<table>
<thead>
<tr>
<th>Object Type</th>
<th>Name</th>
<th>Example</th>
</tr>
</thead>
</table>
| Report      | Z+<Module Area>+R+<Programfunction>+ _ +<meaningful name> | Zhrrr_benefitplans
|             |      | Zfire_vendorinvoice |
| Include     | Z+<Module Area>+<Program Function>+<IncludeType>+ _ +<meaningful name> where, | Zhrie_empbasicpay |
|             |      | Z = Constant Include Type = Data definitions and parameters (D), Error Logging (E), Additional Subroutines (S) |
| Function Group | Z+<Module Area>+F+<meaningful name> Where, | Zmmf_stocktransfer |
|             |      | Z = Constant |
| Class       | Z+<ModuleArea>+Cl+<meaningful name>+<Class definition/implementation> Where, | Zhrcr_xxxxx_def |
|             |      | Z = Constant Cl = Class Class Definition = def Class Implementation = imp |
| Transactions | Z+<Module Area>+<meaningful name> Where, | Zhr_catsaudit |
|             |      | Z = Constant |

Program Function can be
- Conversion (C),
- Data Warehouse (D),
- Enhancement (E),
- Interface (I),
- Report (R),
- Module Program (M)
4.3.1.2 ABAP PROGRAMS

ABAP Program names may be between 5 and 40 characters in length and should be composed of the following subfields:

Z – all programs must begin with the letter “Z”

XX – where XX is the SAP module

Current SAP modules implemented at LACCD are:

- **FI** - Finance
  1. AP - Accounts Payable
  2. AM - Asset Management
  3. GL - General Ledger
  4. GM - Grants Management
  5. FM - Funds Management
  6. MM - Materials Management

- **HR** - Human Resources
  1. TM – Time Management
  2. PY – Payroll
  3. OM – Organizational Management
  4. PM – Personnel Management
    - BEN - Benefits
    - BGO - Budget Management
    - PA - Personnel Administration
    - Retirement Workbench
    - Open Enrollment
  5. Protocol

- **ESS** - Employee Self Services
- **MSS** - Manager Self Services
• LO - Logistics
  1. PM - Plant Maintenance
• PCR/PCS
• SAP BW

- “_” underscore separates the prefix from the remaining program name
- The remaining 36 characters can be as descriptive as required.

When writing a temporary or training program, please use the prefix “ZTST_XX_” for the beginning of your program name. This will indicate that this program should not be used on a regular basis in the production environment.

When making changes to a program, it’s common to make a local object, modify the local object and then copy it back to the original program after initial testing is complete. However, it’s also important to create a transport as soon as you copy the program. If you don’t do this, another programmer can make changes to that program, promote it and then you find the changes wiped out when the local object is copied back. Lock the original object so that another developer does not make changes to the program as well. Do a version compare with object to ensure that another developer does not have any changes that have not yet been promoted to LACCD.
### 4.3.1.3 DATABASE OBJECTS NAMING STANDARDS

<table>
<thead>
<tr>
<th><strong>Object Type</strong></th>
<th><strong>Name</strong></th>
<th><strong>Example</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>Database Table</td>
<td>[Z+\text{&lt;ModuleArea&gt;&gt;+t}_+\text{&lt;meaningful name&gt;}, \text{where}, \text{Z} - \text{Constant}, t - \text{Table}]</td>
<td>\text{Zhrt_employees} \text{Zfit_vendororders}</td>
</tr>
<tr>
<td>Database View</td>
<td>[Z+\text{&lt;ModuleArea&gt;&gt;+v}_+\text{&lt;meaningful name&gt;}, \text{Where}, \text{Z} - \text{Constant}, v - \text{View}]</td>
<td>\text{Zhrv_empbenefitplans} \text{Zfiv_invoices}</td>
</tr>
<tr>
<td>Structure</td>
<td>[Z+\text{&lt;ModuleArea&gt;&gt;+s}_+\text{&lt;meaningful name&gt;}, \text{Where}, \text{Z} - \text{Constant}, s - \text{Structure}]</td>
<td>\text{Zhrs_timemanagement} \text{Zfis_orders}</td>
</tr>
<tr>
<td>Domain</td>
<td>[Z+\text{dm}_+\text{&lt;meaningful name&gt;}, \text{Where, Z - Constant, DM - Domain Name}]</td>
<td>\text{Zdm_ordnr}</td>
</tr>
<tr>
<td>Data Element</td>
<td>[Z+\text{de}_+\text{&lt;meaningful name&gt;}, \text{Where, Z - Constant, de = Data Element}]</td>
<td>\text{Zde_ordnr}</td>
</tr>
<tr>
<td>Lock Object</td>
<td>[E+z+t+_&lt;reference to base table&gt;, \text{Where, T = Table}]</td>
<td>\text{Ezt_orders}</td>
</tr>
<tr>
<td>Search Help</td>
<td>[Z+\text{sh}_+&lt;reference to data that is searched for&gt;, \text{Where, Z = Constant, sh = Search Help}]</td>
<td>\text{Zsh_cust}</td>
</tr>
<tr>
<td>Type Group</td>
<td>[Z+tg+&lt;meaningful name&gt;, \text{Where, Z = Constant, tg = Type Group}]</td>
<td>\text{Ztg_r01}</td>
</tr>
</tbody>
</table>
4.3.1.4 VARIABLE NAMING STANDARDS

<table>
<thead>
<tr>
<th>Variable Type</th>
<th>Prefix</th>
</tr>
</thead>
<tbody>
<tr>
<td>Selection Screen Parameter</td>
<td>p_</td>
</tr>
<tr>
<td>Form routing parameter</td>
<td>P_</td>
</tr>
<tr>
<td>Select-option</td>
<td>s_</td>
</tr>
<tr>
<td>Ranges</td>
<td>r_</td>
</tr>
<tr>
<td>Internal Tables (global)</td>
<td>t_</td>
</tr>
<tr>
<td>Internal Tables (local)</td>
<td>lt_</td>
</tr>
<tr>
<td>Local Structures</td>
<td>ls_</td>
</tr>
<tr>
<td>Global Structures</td>
<td>gs_</td>
</tr>
<tr>
<td>Constants</td>
<td>c_</td>
</tr>
<tr>
<td>Global Constants</td>
<td>gc_</td>
</tr>
<tr>
<td>Local Variables</td>
<td>lv_</td>
</tr>
<tr>
<td>Global Variables</td>
<td>gv_</td>
</tr>
</tbody>
</table>

4.3.1.5 FUNCTION MODULES NAMING STANDARDS

The standard naming convention (ZXX_) should be used for naming Function Modules and Function Groups. The import and export parameters of function modules should be documented at the top of the source code section of the function module with brief descriptions of the fields and their typical contents. Also, any special requirements or usage should be noted.

The first letter of the parameter’s name should indicate the direction in which the parameter was passed:

- Input or importing = i
- Output or exporting = e
- Bi-directional or changing = c
The second letter of the parameter's name should indicate the nature of the formal parameter:

- Single value or variable = v
- Single structure or record (however complicated) = s
- Internal table (however complicated the line structure) = t

Function modules that contain database reads should also contain at least one EXCEPTION parameter.

All RFC's (Remote Function Call) must contain at least one EXCEPTION parameter. When a function module detects an error, it (typically) raises an exception to report the error back to the caller. If you use the ABAP "RAISE" command, the caller gets the number associated with the exception. It is strongly recommended that LACCD-written function modules use the ABAP "RAISE" command.

Similarly the following naming convention has to be used while naming parameters in subroutines and methods.

<table>
<thead>
<tr>
<th>Subroutine</th>
<th>Prefix</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>Using</td>
<td>u</td>
<td>u_matnr</td>
</tr>
<tr>
<td>Changing</td>
<td>c</td>
<td>c_orders</td>
</tr>
<tr>
<td>Tables</td>
<td>t</td>
<td>t_calculating</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Methods</th>
<th>Prefix</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>Importing</td>
<td>i</td>
<td>i_matnr</td>
</tr>
<tr>
<td>Exporting</td>
<td>e</td>
<td>e_suppliers</td>
</tr>
<tr>
<td>Changing</td>
<td>c</td>
<td>c_orders</td>
</tr>
<tr>
<td>Returning</td>
<td>r</td>
<td>r_found_sw1</td>
</tr>
</tbody>
</table>
### 4.3.1.6 WORKFLOW NAMING STANDARDS

<table>
<thead>
<tr>
<th>Object Type</th>
<th>Name</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Business Object</strong></td>
<td>Name can be maximum 10 char length</td>
<td>Zhrbo_newemp</td>
</tr>
<tr>
<td></td>
<td><strong>Z&lt;Module Area&gt;+bo+_&lt;meaningful name&gt;</strong> Where, Z = Constant, bo = Constant</td>
<td>Zfiboveninv</td>
</tr>
<tr>
<td><strong>Workflow</strong></td>
<td>The workflow id is automatically generated. Name can be maximum 12 characters in length</td>
<td>Use abbreviation to describe the workflow. Abbreviation of the workflow should be as descriptive as possible.</td>
</tr>
<tr>
<td><strong>Task</strong></td>
<td>User Abbreviation to describe the task</td>
<td>ZFltd_vendinvc</td>
</tr>
<tr>
<td><strong>Task Id</strong></td>
<td>This is a system generated number. Name can be maximum 12 characters in length</td>
<td></td>
</tr>
</tbody>
</table>
4.3.1.7 WEBDYNPRO ABAP/JAVA APPLICATIONS NAMING STANDARDS

Following are the naming conventions for the Web Dynpro Components:

Plugs

<table>
<thead>
<tr>
<th>Plug Name</th>
<th>Convention</th>
</tr>
</thead>
<tbody>
<tr>
<td>Outbound plug</td>
<td>TO_&lt;to Where&gt;</td>
</tr>
<tr>
<td>Inbound plug</td>
<td>FROM_&lt;from where&gt;</td>
</tr>
</tbody>
</table>

Example

<table>
<thead>
<tr>
<th>View</th>
<th>Properties</th>
<th>Layout</th>
<th>Inbound plugs</th>
<th>Outbound plugs</th>
<th>Context</th>
<th>Attributes</th>
<th>Act</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Outbound Plugs

<table>
<thead>
<tr>
<th>Plug Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>TO_PROGRAM_MAIN</td>
<td>When Next or Prev button is clicked, go to Program_main</td>
</tr>
</tbody>
</table>

UI Elements

Note: <...> means NAME

<table>
<thead>
<tr>
<th>UI Element</th>
<th>Convention</th>
</tr>
</thead>
<tbody>
<tr>
<td>Button</td>
<td>BTN_&lt;...&gt;</td>
</tr>
<tr>
<td>BreadCrumb</td>
<td>BRC_&lt;...&gt;</td>
</tr>
<tr>
<td>ButtonChoice</td>
<td>BTC_&lt;...&gt;</td>
</tr>
<tr>
<td>ButtonRow</td>
<td>BTNRR_&lt;...&gt;</td>
</tr>
<tr>
<td>Caption</td>
<td>CAP_&lt;...&gt;</td>
</tr>
<tr>
<td>CheckBox</td>
<td>CB_&lt;...&gt;</td>
</tr>
<tr>
<td>Checkboxgroup</td>
<td>CBG_&lt;...&gt;</td>
</tr>
<tr>
<td>DateNavigator</td>
<td>DTN_&lt;...&gt;</td>
</tr>
<tr>
<td>UI Element</td>
<td>Convention</td>
</tr>
<tr>
<td>----------------------------------</td>
<td>--------------</td>
</tr>
<tr>
<td>DropDownByIndex</td>
<td>DDBI_&lt;...&gt;</td>
</tr>
<tr>
<td>DropDownByKey</td>
<td>DDBK_&lt;...&gt;</td>
</tr>
<tr>
<td>FormattedTextView</td>
<td>FTV_&lt;...&gt;</td>
</tr>
<tr>
<td>FileDownload</td>
<td>FD_&lt;...&gt;</td>
</tr>
<tr>
<td>FileUpload</td>
<td>FU_&lt;...&gt;</td>
</tr>
<tr>
<td>Group</td>
<td>GR_&lt;...&gt;</td>
</tr>
<tr>
<td>HorizontalGutter</td>
<td>HG_&lt;...&gt;</td>
</tr>
<tr>
<td>Inputfield</td>
<td>INP_&lt;...&gt;</td>
</tr>
<tr>
<td>ItemListBox</td>
<td>ILB_&lt;...&gt;</td>
</tr>
<tr>
<td>Image</td>
<td>IMG_&lt;...&gt;</td>
</tr>
<tr>
<td>InvisibleElement</td>
<td>IVE_&lt;...&gt;</td>
</tr>
<tr>
<td>Label</td>
<td>LBL_&lt;...&gt;</td>
</tr>
<tr>
<td>LinkToAction</td>
<td>LTA_&lt;...&gt;</td>
</tr>
<tr>
<td>LinkToURL</td>
<td>LTURL_&lt;...&gt;</td>
</tr>
<tr>
<td>MessageArea</td>
<td>MSGA_&lt;...&gt;</td>
</tr>
<tr>
<td>PageHeader</td>
<td>PH_&lt;...&gt;</td>
</tr>
<tr>
<td>RadioButton</td>
<td>RB_&lt;...&gt;</td>
</tr>
<tr>
<td>RadioButtonIndex</td>
<td>RBGI_&lt;...&gt;</td>
</tr>
<tr>
<td>RadioButtonGroupByKey</td>
<td>RBGK_&lt;...&gt;</td>
</tr>
<tr>
<td>RoadMap</td>
<td>RM_&lt;...&gt;</td>
</tr>
<tr>
<td>ScrollContainer</td>
<td>SC_&lt;...&gt;</td>
</tr>
<tr>
<td>Tab</td>
<td>T_&lt;...&gt;</td>
</tr>
<tr>
<td>Table</td>
<td>TBL_&lt;...&gt;</td>
</tr>
<tr>
<td>Tableheader</td>
<td>TBLH_&lt;...&gt;</td>
</tr>
<tr>
<td>TableColumn</td>
<td>TBLC_&lt;...&gt;</td>
</tr>
<tr>
<td>TableCellEditor</td>
<td>TBLCE_&lt;...&gt;</td>
</tr>
<tr>
<td>TableColumnHeader</td>
<td>TBLCH_&lt;...&gt;</td>
</tr>
<tr>
<td>TabStrip</td>
<td>TS_&lt;...&gt;</td>
</tr>
<tr>
<td>TransparentContainer</td>
<td>TC_&lt;...&gt;</td>
</tr>
<tr>
<td>TextEdit</td>
<td>TXTE_&lt;...&gt;</td>
</tr>
</tbody>
</table>
UI Element | Convention
---|---
TextView | TXTV_<...>
Toolbar | TB_<...>
Tree | TR_<...>
TreeItemType | TIT_<...>
TreeNodeType | TNT_<...>
ViewContainerElement | VC_<...>

### Example For Button UI

![Example Image]

**Web Dynpro Code**

Use the same naming convention Web Dynpro Code Wizard generates. For example,

lo_nd_<context name>
lo_el_<context name>
ls_<context_name>
lt_<context name>
lv_<descriptive name>

See the below sample code for variable names, method call etc., that Web Dynpro Code Wizard generates.

```abap
DATA lo_nd_node_po_view TYPE REF TO if_wd_context_node.
DATA lo_el_node_po_view TYPE REF TO if_wd_context_element.
DATA is_node_po_view TYPE wd_this->element_node_po_view.

wd_this->adjust_view_area( 
  i_index = 
  "integer"
).

DATA lo_interfacecontroller TYPE REF TO iwc1_salv_wd_table .
lo_interfacecontroller = wd_this->wd_cp1fc_salv_po_create( ).

DATA lo_value TYPE REF TO c1_salv_wd_config_table.
lo_value = lo_interfacecontroller->get_model( ).
```

The standard of variable names which are not generated by Web Dynpro Code Wizard should be the same as regular ABAP variable standards.

lo_<descriptive name> ; local object
go_< descriptive name> ; global object
go_< descriptive name> ; global pointer to object
gv_< descriptive name> ; global variable
gs_< descriptive name> ; global structure
gt_< descriptive name> ; global table

See Checklist for High Performance WDA Programming

http://help.sap.com/saphelp_nw70ehp1/helpdata/en/5e/b29046859d48d68af26c16c75d4a89/content.htm

See SAP recommended Web Dynpro Naming Conventions

4.3.2 DOCUMENTATION STANDARDS

Software documentation exists in two forms, external and internal. External documentation is maintained outside of the source code, such as specifications, help files, and design documents.

Internal documentation is composed of comments that developers should write within the source code at development time.

**Following are the recommended commenting techniques:**

Comment Content: Comment implementations meaningfully in such a way that the comments describe *why* something is done and not *how*.

- Efforts should be made to update and maintain comments in parallel with the source code.
- Avoid adding comments at the end of a line of code; end-line comments make code more difficult to read. However, end-line comments are appropriate when annotating variable declarations. In this case, align all end-line comments at a common tab stop.
- Throughout the application, construct comments using a uniform style, with consistent punctuation and structure.
- Also, include comments at top of the source code for the following:
  - When creating a new program, record the Author and date for future reference or consultation
  - When making changes, the developer name who made the change, the transport number and the date
  - Comment with transaction code, menu path... (Optional)

If necessary, documentation notes of some technical aspects of the program.
In the source codes:
- Make a brief comment for each block of codes including name, date, issue log # transport # and description.
- Comment with the transport number the lines being changed

The idea is to keep high level (general for user) documentation in the ‘Documentation’ section and the technical/detail in the source. The documentation in the ‘Documentation’ section can be extracted from the specifications document. This documentation can refer more to business processes.

When modifying ABAP programs include the change request number and the transport number(s) in the opening comments of the source code. Including the change request number means that you don’t have to write a lengthy explanation of the change, also it will enable the user to trace back to the original requirements/specs when questions arise.

A template program ZL_TEMP in DEV client 300 has been provided for your reference

Documentation template:

```
*---------------------------------------------------------------------* *
* & Program Name :                                                 * *
* & Developer    :                                                 * *
* & Date        :                                                 * *
* & Issue Log Number :                                            * *
* & Description  :                                                 * *
* & Transaction Code :                                             * *
* & Z table      :                                                 * *
* & Batch Job Name :                                               * *
* & Logical Database :                                             * *
* & SAPscript Form :                                               * *
* & SAPscript Standard Text :                                      * *
*---------------------------------------------------------------------* *
* & MODIFICATION HISTORY :                                         * *
*---------------------------------------------------------------------* *
* & Name   Date    IL#   Transport #   Description                  * *
* & xxxxx 03/27/09 1111   DEVK921471 xxxxx xxxxxxxxxxxxxxxxxxxxxxx x xxxxx * *
```
ABAP Code should be formatted in a consistent and organized manner.

- Standardize formatting should be used to format all programs, function modules, etc. To setup standard formatting select ‘Setting’ under the ‘Utilities’ option on the SAP toolbar. Select ‘ABAP Editor’ option. Within this option, select ‘Pretty Printer’.

Check the following options:

- Indent
- Convert Uppercase/lowercase
- Keyword Uppercase

- One command per line - Each ABAP/4 command consists of a sentence ending with a period. As a standard, start each new command on a new line. This will allow for easier deleting, commenting, and debugging.
• **Indented source code** - For command statements that have a corresponding “END” (such as IF...ENDIF, SELECT...ENDSELECT, LOOP...ENDLOOP), the “END” part of the statement should be placed in the same column as the beginning part of the statement. It is helpful to also comment the “END” statement by using the “parameter on the same line.

For example,

```
LOOP at ITAB
       .......... Statements here
END LOOP   "ITAB table
```

### 4.4 PROGRAMMING PRACTICES

Employing programming best practices result in programs that are correct, robust, well structured, readable and maintainable.

During the development of a brand new program the below mentioned order should be followed:

- Tables
- Parameters & Select-options
- Data declarations
- Constants
- Initializations
- Selection screen verifications
- Mainline
- Page events
- Navigation logic (user commands, line selections)
- Forms
The developer should maintain the same order of the sections across the programs (i.e. data declarations and parameters come before events and events before forms).

In the data declaration section, group internal tables together, range together and individual fields together. Events should only use performs – no statements.

### 4.4.1 REUSABLE CODE

If a block of code is executed more than once, it should be placed in a subroutine at the bottom of the code or in a function module. This makes the code more readable, requires less indentation, and is easier to debug.

### 4.4.2 PARAMETER PASSING IN SUBROUTINE

Whenever possible use a TYPE or LIKE statement when specifying the formal parameters of a subroutine. This is a good programming style, plus it allows the ABAP compiler to generate more efficient code (which can increase performance up to a factor of 2x).

### 4.4.3 TEXT HANDLING

Variable names should not be used on the parameter selection screen for any program placed in production. Use the “Text Element” function found in the ABAP editor to describe the selection criteria and relate it to the variable name.

INCLUDE files can't define their own Text Elements - any Text Elements to which they refer must be defined in the main program which invokes the INCLUDE file.

In cases other than INCLUDE files, constant text which is printed in a report can be stored as Text Symbols.

There are two ways that you can code references to these Text Symbols, either using
**TEXT-xxx or using '\(\ldots\)'(xxx).** Here, xxx stands for a 3-digit number, and ... for the text of the Text Symbol.

The first form requires that you separately define a Text Symbol for number xxx. If xxx isn’t a defined Text Symbol, the output is empty. The second form improves the readability of the program. The text between the single quotes should correspond to the text stored as the value of the Text Symbol. If it does not, the system uses the text stored for the Text Symbol. Exception: If there is no text saved under number xxx, the text between the single quotes is used.

Example: Text symbol number 001 has the text 'Please enter your name'. The following commands all have the same output: "Please enter your name":

```abap
WRITE: /TEXT-001
   '/'Please enter your name'\(001\),
   '/'What is your name? '\(001\).
```

In the ABAP Editor, you can compare the texts used in the program with the texts stored as Text Symbols by choosing "Goto -> Text elements -> Text symbols", then "Utilities -> Adjust -> Text symbols", then selecting the "Text symbols defined repeatedly/differently in program" radio button and clicking on the "Edit" soft button.

The advantages to the \(\ldots\)'(xxx) form are readability and that the Text Symbol only needs to be maintained for multi-lingual clients (for those installations which use multiple languages). The advantage to the TEXT-xxx form is easier maintainability if TEXT-xxx is coded several times in the program and it needs to be changed.

**TIP:** You can use the INITIALIZATION event of the “include” program to set the values of these text elements.
4.4.4 USERIDS IN PROGRAMS

In no case should a production program or function contain a UserID as either literal or constant data. In the development system it may be necessary to code temporary breakpoints for specific UserID’s, however; these debugging techniques must be removed before the program is transported.

4.4.5 MESSAGES

Messages are texts that are created using the message maintenance (Transaction SE91) and are stored in the T100 system table.

Declare the message class in the report statement. While it is possible to specify the message class each time a message is output, it is easier to specify once it in the report statement. You can still use a message from another class than the one defined in the report by adding the class in parentheses after the message.

The following message types are distinguished:

- Status Message (S)
- Information Message (I)
- Warning (W)
- Error Message (E)
- Termination Message (A)

Messages should use proper grammar and punctuation.
4.4.6 DATA DICTIONARY

While making changes or updates to the Data Dictionary, the following points should be kept in mind.

- Tables should have ‘create’ and ‘change date/user/time’ fields.
- Z* tables should be created using data class USER1 or USER2.
- Create lock objects for tables that will be updated by users.
- Perform impact analysis for any field, data element or domain changes.
- Utilize include structures for tables and structures used in screens. This will minimize maintenance requirements.
- Avoid buffering the full table, since it takes up system memory.
- Consult with the Basis team on data dictionary changes and update the data model if applicable.
- Consider using existing data elements and domains when creating new fields.
- Insert the client field (mandt) in all tables and indexes.
- Insert check tables if applicable (foreign key relationship) when inserting fields in a table.
- Consider whether or not to initialize new fields. Fields not initialized will contain nulls and initializing a field will trigger a rebuild of the table.
- Insert documentation for data elements.
- When creating new data elements, consider if creating parameter ids will be useful and if change documents are required.
- Determine if creating a table maintenance view will suffice when accessing the table (instead of creating a new screen).
- Determine if creating a view will optimize database accesses.

When creating a new table or field, consider what the best techniques are for presenting pick lists and which technique will promote consistency in presentation to the user (e.g. Search helps vs. Programmed help lists).
4.4.7 HP-UX/SERVER FILES

When designing the program keep in mind the following: use valid UNIX file names. Generally speaking, the user should not be allowed to enter the directory names, only the file name itself. New programs should be following the requirement: List UNIX directories, their retention period, when they should be used, by which application.

**Calling UNIX commands and scripts from ABAP Programs**

Often, it is necessary to execute an external program as part of an ABAP program. However, user should be very careful in doing so because any external command is executed as the UNIX user <SID>adm. For example, if a program is written to call the command "rm" on the development system it is executed at the system level as the user "DEVadm".

For an example of how to call UNIX system commands, look at program ZUNIXCMD on system DEV. Remember, you can call external programs either synchronously so that your ABAP program will wait until they complete before continuing, or you can call them asynchronously so that your ABAP program will continue execution immediately.

Earlier programs written at LACCD used the "CALL SYSTEM" command. In the future, this command should not be included in any new programs and should be removed from existing programs when they are revised.

4.4.8 PARAMETERS AND SELECTION CRITERIA

When designing the selection screen, the Developer should assign the following attributes correctly to the selection screen parameters:

- Default values
- Search help
- PIDs
4.4.9 ONLINE PROGRAMMING

- Ensure consistency in the interface. If applicable, insert the area menu as includes in the new menu.
- Screens should be user-friendly, intuitive and consistent with both Windows and SAP standards.
- Standardize function key (icon) on GUI Status across screens in different modules.
- Display descriptions of coded values on the screen. The coded values may be suppressed when in display mode in certain situations and it will enhance the user-friendliness of the screen.
- Suppress from view any functions that are not available to the user based on their security access. This will enhance the user-friendliness of the screen.
- Ensure consistency with error messages. Standardize when to use information message (I), successful message (S), etc.
- Data that can be updated should have screens with display, change, add and delete modes, with display mode as the default mode. Create appropriate security objects to correspond with each mode.
- Use table locks for any database updates.
- Create a prototype if it is possible and elicit feedback from both users and the team.
- Provide confirmation prompts when deleting records. When deleting a row, consider if the record should be marked as ‘deleted’ or physically deleted from the table.
- Provide a warning if the user is in change mode and backs out without saving any changes first (‘are you sure?’). Track if any fields are changed when in change mode.
- Ensure that new screens can be accessed via BDC mode.

The Developer is responsible for ensuring test variants are not transported to the production instance.
4.4.10 OPTIMIZATION CONSIDERATIONS

- Use SELECT * with care. Do not use SELECT * when populating internal tables if NOT all columns are required; only select the necessary columns.
- Don’t use columns with low/no selectivity in a where clause.
- Don’t index columns with low/no selectivity. Selectivity refers to the number of distinct values.
- To reduce database access by repeated selections, it’s better to scan a table once into the internal table and then read the internal table using statement READ TABLE ... BINARY SEARCH. But be careful and consider resource issues when selecting into an internal table – balance between optimization and memory resources.
- Avoid MOVE-CORRESPONDING statements. Instead use MOVE statement and move fields individually or store MOVE statements in the FORM.
- Check SAP performance tips program online (transaction SE30)
- Structure of internal table should match the order the fields are returned from the select statement when selecting into an internal table thereby avoiding usage of the statement ‘into corresponding fields’.
- Avoid nested selects if possible.
- Be careful using ‘CHECK’ statements, consider incorporating the selection criteria into the select statement.
- Use the SQL trace and runtime analysis to check the program.
4.4.11 SAPSCRIPT VS SAP SMART FORMS

Try to use text elements as include texts in the form especially if changes in the text are likely to be made somewhat frequently. It may be more feasible that some local text elements can be modified easily by users.

Given below are the differences between SAP Smartforms and SAPScripts:

- Multiple page formats are possible in Smartforms which is not the case in SAPScripts.
- It is possible to have a Smartform without a main window.
- Labels cannot be created in Smartforms.
- Routines can be written in Smartforms tool.
- Smartforms generates a function module when activated.

4.4.12 PRODUCTION DOCUMENTATION

When moving the program to the production instance, the Developer must update Production documentation (Documentation Project in QC/ZPRODDOC) with specific instructions including the new job name, when to run it, dependencies, etc.

4.4.13 SECURITY

- Consider if the new security object should be transaction or object based.
- Determine if access should be table or field level security.
- Set levels to represent the various modes of access (e.g. display, change, add and delete).

Change any existing profiles if required.
4.4.14 AREA MENUS

To make it easier for end users to find a report or a function, they are usually organized in the menus by means of area menus or reporting trees. Business specifications should provide developer information on how a report or a function will be accessed. However, it is the responsibility of a developer to solicit the information if it is not provided.

Area menus and reporting trees are controlled by the Coordinators. Therefore a developer should send a request (i.e., e-mail) to them asking to add a new report to the menu, providing them transaction code, navigation path and description (if new or changing).

4.4.15 SAP QUERY

The SAP Query application is used to create reports not already contained in the default. It has been designed for users with little or no knowledge of the SAP programming language ABAP.

SAP Query offers users a broad range of ways to define reports and create different types of reports such as basic lists, statistics, and ranked lists.

The SAP Query comprises five components: Queries, InfoSet Query, InfoSets, User Groups and Translation/Query.

Classic reporting – the creation of lists, statistics and ranked lists – are covered by the InfoSet Query and Queries components. Other components’ range of functions cover the maintenance of InfoSets, the administration of user groups and also the translation of texts created in the SAP Query.

All data required by a user for a report can be read from various tables.
4.4.16 USER EXITS

User exits (Function module exits) are exits developed by SAP. The exit is implemented as a call to a function module. The code for the function module is written by the developer. You are not writing the code directly in the function module, but in the Include that is implemented in the function module.

The naming standard of function modules for function exits is:

```
EXIT_<program name><3 digit suffix>
```

The call to a function exit is implemented as:

```
CALL CUSTOMER-FUNCTION <3 digit suffix>
```

**Example:**

```
CALL CUSTOMER-FUNCTION '003'
  exporting
    xvbak    = vbak
    xvbuk    = vbuk
    xkomk    = tkomk
  importing
    lvf_subrc = lvf_subrc
  tables
    xvbfa = xvbfa
    xvbap = xvbap
    xvbup = xvbup
```

The exit calls function module `EXIT_SAPMV45A_003`

**To Find User Exists** - Display the program where you are searching for the exit and search for **CALL CUSTOMER-EXIT**. If you know the Exit name, go to transaction CMOD. Choose menu Utilities->SAP Enhancements. Enter the exit name and press enter.

You will now come to a screen that shows the function exits for the exit.
Example:

```abap
MODULE user_exit_0001 INPUT
  CASE okcode.
    WHEN 'BACK OR EXIT'.
      CASE sy-dynnr.
        WHEN '100'.
          SET SCREEN 0.
          LEAVE SCREEN.
        WHEN '200'.
          *** Note that you can write any code that satisfies your needs. But in this case, this was written as a sample code for reference sake ***
          SET SCREEN 100.
          LEAVE SCREEN.
        ENDCASE.
    END_CASE.
  END_CASE.
ENDMODULE.
```

---

4.4.17 BADI (BUSINESS ADD-INS)

SAP Standard Programs can be easily modified/changed using Business Add-Ins or BADIs. And all this can be done without any system level modifications.

BADIs are based on ABAP objects and are new techniques introduced by SAP for changing the SAP Standard programs as per the user requirements. The concept is similar to User Exits but BADI’s make use of ABAP Objects. Many industries have some specific requirements that may not be configurable in SAP. This can be easily achieved using BADIs. The original Object does not change as this piece of code is inserted in specific points using BADIs.

It is recommended to use BADIs over USER-EXIT because,

1. BADIs can be used any number of times, where as USER-EXITS can be used only one time
Ex:- if you are assigning a USER-EXIT to a project in (CMOD), then you cannot assign the same to other project.

2. BADIs are OOPS based.

3. A single Business Add-In contains all of the interfaces necessary to implement a specific task.

4. BADIs can be defined according to filter values. This allows you to control add-in implementation and make it dependent on specific criteria.

4.4.18 AUTHORIZATION OBJECTS

An authorization object groups together 1 to 10 authorization fields which can then be checked as a combination. The programmer can create authorization fields by selecting Tools → ABAP Workbench → Development → Other tools → Authorization obj → Objects.

Example: The authorization objects S_TRVL_BKS groups together the authorization fields ACTVT and CUSTTYPE.

In an ABAP program, there are no automatic authorization checks associated with Open SQL statements. This can cause problems, since Open SQL and Native SQL statements allow unrestricted access to all database tables.

An ABAP program must have a Transaction Code Authorization object (S_TCODE). Other Authorization Objects should also be included. To check the authorization of the user of an ABAP program, use the AUTHORITY-CHECK statement:

```
AUTHORITY-CHECK OBJECT '<object>'
    ID '<name1>'  FIELD <f1>
    ID '<name2>'  FIELD <f2>
    ..............
    ID '<name10>'  FIELD <f10>.
```
Eg: T-Code authorization

Check Authorization

```
AUTHORITY-CHECK OBJECT 'S_TCODE'
   ID 'TCD' FIELD 'ZHRPY_EMAIL_PAYSTUB'.
   IF sy-subrc <> 0.
      MESSAGE e000(zhr) WITH 'You need to have'(i04)
         'Transaction authorization to execute.'(i05)
         'Contact System Administrator.'(i06)
         'Tcode: ZHRPY_EMAIL_PAYSTUB'(i07).
   ENDIF.
```

---

### 4.4.19 Declaring Internal Tables Using R/3 Release 4.X Syntax

R/3 release 4.6 contains capabilities for hashing and sorting internal tables that improve system performance. The 4.6 syntax should be used whenever coding internal tables. The following examples illustrate the use of this syntax.

First, declare the header line or work area (wa) as a type:

```
TYPES: begin of <line_type>,
   field1 like ..., 
   ....
   Fieldn like ..,
   end of <line_type>
```

For example, here is a table line type with some vendor fields

```
TYPES: Begin of TY_VENDOR_DATA,
   LIFNR type LFA1-LIFNR,
   NAME1 type LFA1-NAM1,
   End of TY_VENDOR_DATA.
```

Or

```
TYPES: <line_type> like <DD table or ddstructure>.
```

This version would include ALL of the vendor (LFA1) fields

```
TYPES: TY_VENDOR_DATA TYPE LFA1.
```
Declare the internal table (see below) or, optionally, declare a table type. Declaring a table type would be preferable in those cases where your program declares several internal tables of the same type (and therefore the table type can be reused by each internal table declaration) or where the internal table is passed to subroutines or methods (and therefore the table type can be included in the form/method interface definition.

\[
\text{TYPES: } <\text{t_table_type}> \text{ type } <\text{line_type}> \text{ occurs 0.}
\]

The example below defines a standard (i.e. Unsorted) table type

\[
\text{TYPES: } \text{ty}_t\text{_vendor} \text{ type } \text{ty}_\text{vendor}_\text{data} \text{ occurs 0.}
\]

Or

\[
\text{TYPES: } <\text{t_table_type}> \text{ type}
\]
\[
\quad [\text{sorted}|\text{hashed}|\text{standard}|\text{index}|\text{any}]
\]
\[
\quad \text{table of } <\text{line_type}>
\]
\[
\quad \text{with } [\text{unique}|\text{non-unique}] \text{ key } <\text{key definition}>].
\]

The example below defines a vendor table type sorted by vendor number

\[
\text{TYPES: } \text{ty}_t\text{_vendor} \text{ type } \text{sorted table of } \text{ty}_\text{vendor}_\text{data}
\]
\[
\quad \text{with } \text{unique key lifnr}.
\]

Finally declare the table and the work area:

\[
\text{DATA: } <\text{t_table}> \text{ type } <\text{t_table_type}>, "\text{itab w/ no header line}
\]
\[
\text{DATA: } <\text{wa_table}> \text{ type } <\text{line_type}>, "\text{work area for t_table}
\]

The example below defines an itab using the table type defined above (Note: depending on which of the above two table types are choosen, will define either a standard or sorted internal table.)

\[
\text{DATA: } \text{t}\_\text{vendor} \text{ type } \text{ty}_t\text{_vendor}, "\text{itab has no header line}
\]
\[
\quad \text{wa}\_\text{vendor} \text{ type } \text{ty}_\text{vendor}_\text{data}, "\text{work area for t_vendor}
\]

Declaration of an internal table using a table line type
**4. ABAP Development**

1. **Module Pool**
   - Transactions are maintained using SAP transaction SE38 (ABAP Editor).
   - Naming convention – SAPMZXX_
   - Naming convention for includes
      - i. MZXX_...TOP – Global data
      - ii. MZXX_...O01 – PBO modules
      - iii. MZXX_...I01 – PAI modules
      - iv. MZXX_...F01 – Form routines

2. **Screens**
   
   Transactions are maintained using SAP transaction SE51 (Screen Painter).

**DATA:**

```abap
DATA: <t_table> type
    [sorted|hashed|standard|index|any] table
    of <line_type>
    with [unique|non-unique] key <key definition>]
    [with header_line].
```

This example below declares a standard, unsorted internal table

```abap
DATA: t_vendor type ty_vendor_data occurs 0 with header line.
```

This example declares an internal table sorted by vendor number

```abap
DATA: t_vendor type sorted table of ty_vendor_data
    with unique key lifnr.
    With header line.
```

4.5 **DIALOG PROGRAM STANDARDS**

---

**LACCD| Confidential Version 2.0 - 2013**
3. GUI Status

Standard menu bar, application toolbar and standard toolbar are maintained using SAP transaction SE41 (Menu Painter).

4. Miscellaneous

- Initialize all screen fields and global variables in the PBO (Process before Output) module.
- PAI (Process after Input) should contain the logic to be executed after the user has selected a function key, menu item, etc.
- POV (Process on Value-Request) should contain logic to display list of possible values on F4 request.
- POH (Process on Help-Request) should contain logic to display help information on F1 request.

4.6 TRANSACTION STANDARDS

Reports and Dialog Programs

- Transactions are maintained using SAP transaction SE93 (Maintain Transactions).
- Transaction Code name can be up to 20 characters. The standard naming convention (ZXX_) should be used (see Naming Standards).
- The standard package is always ZDEV.
- GUI Support – select all (HTML, Java, and Windows).
4. ABAP Development

LACCD/ Confidential Version 2.0 - 2013
4.7 MAKING PROGRAM AND FUNCTION MODULES OBSOLETE

4.7.1 STEPS TO MAKE A PROGRAM OBSOLETE

1. Verify that the program is not referenced by any other programs, transactions or dialog modules that are not obsolete in either the production or development environments. If it is, notify the person who requested the program to be made obsolete and stop here.

2. Reconcile the development version with the production version, and revert development to the same version as production if it is different. (Use the Version Management function to do this). Make sure that the retrieved version can be generated without errors. If there are errors - return back to the original active version.

3. If you have to go back to the previous version, make sure that any active development of an object in unreleased requests/transport is saved by releasing the request/transport.

4. If the program is not a function module, go to transaction SE38, and put in the program name which is to be made obsolete. Click on Attributes, and then click on "Change". Change the title of the program to start with the word "Obsolete". Change the Authorization Group to "ZINACTVO" (the last character is the digit ZERO), and the Application to "* Cross-Application". Attempt to change the Development Class to "ZZZ0" (the last character is the digit ZERO). If you have problems changing the development class, go back to the ABAP editor initial screen. Click on "Goto" at the top of the screen, and then click on "Object Directory entry" on the dropdown. Make the change and save.

5. Send all the information about the program (including by whom & when it was determined that this program is obsolete) to the SAP ERP Manager. Please be sure to send your e-mail before releasing the program.
6. Transport the changes made to QAS and Production.

4.7.2 STEPS TO MAKE FUNCTION MODULE OBSOLETE

Since SAP doesn't let us change the authorization group for one function module, we (LACCD) are using the following six steps in place of step 4 above to make a function module obsolete:

1. Under ATTRIBUTES tab, change the short text (i.e. One line description) of the function module to start with the word "Obsolete".

2. Use the editor function to make all the source code lines of the function module into comments. (On the PC version of the ABAP editor, you can use Control-A to select all the code and Control-< to make all the selected code into comments. Then select the menu item Utilities -> Block/buffer -> Insert comment* To make all the selected code into comments.)

   * NOTE: you cannot comment out the key lines Function/Endfunction.

3. Insert a comment at the top of the function module source code stating that the function module is now obsolete.

4. Put the following ABAP statement in the source code in order to force a short dump should someone try to execute the function module:

   ```abap
   MESSAGE ID 'ZZ' TYPE 'X' NUMBER '132' WITH 'Function module'
   '<name of function module>'.
   ```

5. Use the ABAP Workbench command to mark the function module as obsolete. (Go to SE37, type in the function module name and click on the "change" button. Go to the "Attributes" tab. Select the menu item:

   ```abap
   Function module -> Release -> Object obsolete
   ```

   Given below is an example showing steps 2,3, and 4:

   ```abap
   FUNCTION Z_INSERT_TR_ACCOUNT .
   ****** This function module is OBSOLETE **********
   ```
**MESSAGE ID 'ZZ' TYPE 'X' NUMBER '132' WITH**
'Function module' 'Z_INSERT_TR_ACCOUNT'.

**""""----------------------------------------------""""
**""""Update function module:
**""**""Local interface:
**"** IMPORTING
**"** VALUE(TRAV_ACC) LIKE ZTRV1 STRUCTURE ZTRV1
**"** EXCEPTIONS
**"** INSERT_ERROR
**""""----------------------------------------------
**
* MOVE-CORRESPONDING TRAV_ACC TO ZTRV1.
* INSERT ZTRV1.
* IF SY-SUBRC <> 0.
* RAISE INSERT_ERROR.
* ENDIF.
ENDFUNCTION.

### 4.8 SELECT-OPTIONS IN ABAP PROGRAMS

Sometimes it is necessary to restrict the user from giving a broad range of data selection capabilities. The syntax of the select-option declaration allows a certain measure of control over this. The NO INTERVALS addition will remove the user’s ability to specify ranges of values to include or exclude while the NO-EXTENSION addition removes the user’s ability to specify more than one line of selection criteria. To impose any other restrictions (say, to prevent the use of patterns) an SAP-provided function module named SELECT_OPTIONS_RESTRICT can be used.

The function module SELECT_OPTIONS_RESTRICT restricts the capability of one or more select-options for a given program based on the restriction criteria passed to it in a structure of the form SSCR_RESTRICT. This structure consists of two internal tables. The first (having structure SSCR_ASS) holds the names of those select-options we wish to
control. The second (having structure SSCR_OPT_LIST) specifies exactly which capabilities are allowed or disallowed for each of those select-options named in the first internal table.

This technique is employed within a standard LACCD include file in order to restrict the capabilities of the company code select-option. As a result of this function call, the user is allowed only to provide explicit company code values. That is, no patterns, relational operators or exclusions are allowed.

It is important to note that the SELECT_OPTIONS_RESTRICT function module may only be called once during a given program run. Therefore, if you wish to restrict the capabilities of more than one of a given program's select-options, you must load all of the desired restrictions for all of the affected select-options into the SSCR_RESTRICT structure prior to the call to SELECT_OPTIONS_RESTRICT.

### 4.9 UPLOADING AND DOWNLOADING FILES

In R/3 4.6C, ABAP programs should use the WS_UPLOAD and WS_DOWNLOAD function modules (rather than the GUI_UPLOAD and GUI_DOWNLOAD function modules or static methods) to upload/download files between the SAP GUI and R/3 system.

In R/3 6.20 and later, the WS_UPLOAD and WS_DOWNLOAD function modules are still available, but the GUI_UPLOAD and GUI_DOWNLOAD static methods of the CL_GUI_FRONTEND_SERVICES class are preferred.
4.10 BATCH DATA COMMUNICATION (BDC)

4.10.1 FORMATTING DATE FIELDS TO BE COMPATIBLE WITH USER PREFERENCE

When writing BDCs that read dates to be entered into SAP transactions (either by generating batch input sessions or by using call transaction) problems occur when determining what date format to use when entering the data into its corresponding field. Assume that the ABAP program that must call the transaction (or generates the batch input session) can read the 'date' field properly from the input file, but it does not know which format the data entry screen of the transaction requires as this may change based on the user's setup defaults.

4.10.2 FORMATTING A NUMBER TO BE COMPATIBLE WITH USER PREFERENCE

When writing BDCs that read numbers to be entered into SAP transactions (either by generating batch input sessions or by using call transaction) problems occur when determining what number format to use when entering the data into its corresponding field. Assume that the ABAP program that must call the transaction (or generates the batch input session) can read the 'number' field properly from the input file, but it does not know which format the data entry screen of the transaction requires as this may change based on the user’s setup defaults.
5. PERFORMANCE STANDARDS

5.1 GENERAL PERFORMANCE STANDARDS:

All ABAP programs should be developed using the most efficient means possible. Program efficiency guidelines are included in the EBS ABAP Programming Standards.

Some of the factors impacting application performance that are of particular concern for ABAP programmers are:

- Program structure
- Programming language commands / syntax
- SQL statements
- Input/Output operations

- **"Dead" code** - Avoid leaving "dead" code in the program. Comment out (or delete) variables that are not referenced and code that is not executed. Use program --> check --> extended program to check to see a list of variables which are not referenced statically.

- **Use Logical Databases** - Choose the most efficient logical data base possible. Study the selection criteria and which secondary indexes are used for that view. Provide the appropriate selection criteria to limit the number of data base reads. Force users to provide selection criteria by evaluating the selection criteria entered on the selection screen during the AT SELECTION-SCREEN event. Finally, when possible take advantage of the match codes to increase speed.

- **Subroutine usage** - For good modularization, the decision of whether or not to execute a subroutine should be made before the subroutine is called. For example:
  - This is better:
    - IF f1 NE 0.
    - PERFORM sub1.
    - ENDF.
5. Performance Standards

- **IF statements** - When coding IF tests, nest the testing conditions so that the outer conditions are those which are most likely to fail. For logical expressions with AND, place the mostly likely false first and for the OR, place the mostly likely true first.

- **CASE vs. Nested Ifs** - When testing fields "equal to" something, one can use either the nested IF or the CASE statement. The CASE is better for two reasons. It is easier to read and after about five nested IF’s the performance of the CASE is more efficient.

- **MOVE-ing structures** - When records a and b have the exact same structure, it is more efficient to MOVE a TO b than to MOVE-CORRESPONDING a TO b.

  - MOVE BSEG TO *BSEG. is better than
  - MOVE-CORRESPONDING BSEG TO *BSEG.

- **SELECT and SELECT SINGLE** - When using the SELECT statement, study the key and always provide as much of the left-most part of the key as possible. If the entire key can be qualified, code a SELECT SINGLE not just a SELECT. If you are only interested in the first row or there is only one row to be returned, using SELECT SINGLE can increase performance by up to three times.

- **Small internal tables vs. Complete internal tables** - In general it is better to minimize the number of fields declared in an internal table. While it may be convenient to declare an internal table using the LIKE command, in most cases, programs will not use all fields in the SAP standard table.

  - For example, use this:
    - DATA: begin of t_vbak occurs 0,
    - Vbeln like vbak-vbeln,
End of t_vbak.

Instead of this:

Data:  t_vbak like vbak occurs 0 with header line.

- **Row-level processing of a table** - Selecting data into an internal table using an array fetch versus a SELECT-ENDELECT loop will give at least a 2x performance improvement. After the data has been put into the internal table, then row-level processing can be done.

- For example, Use:
  - Into <itab> (corresponding fields of itab)
  - Where ...
  - Loop at <itab>
  - <do the row level processing here>
  - Endloop.

- Instead of using
  - Select... from table <...
  - Where ...
  - <row-level processing>
  - Append<itab>
  - Endselect

- **Row-level processing and SELECT SINGLE** - Similar to the processing of a SELECT-ENDSELECT loop, when calling multiple SELECT-SINGLE commands on a non-buffered table (check Data Dictionary -> Technical Info), do the following to improve performance:

  - Use the SELECT into <itab> to buffer the necessary rows in an internal table, then
  - Sort the rows by the key fields, then
  - Use a READ TABLE WITH KEY ... BINARY SEARCH in place of the SELECT SINGLE command. Note that this only make sense when the table you are buffering is not too large (this decision must be made on a case by case basis).

- **Reading single records of internal tables** - When reading a single record in an internal table, the READ TABLE WITH KEY is not a direct READ. This means that if the data is not sorted according to the key, the system must sequentially read the table. Therefore, you should:
- Sort the table

- **Use** **READ TABLE WITH KEY BINARY SEARCH** for better performance.

- **Sorting internal tables** - When sorting internal tables, specify the fields to sorted.
  - **SORT ITAB BY FLD1 FLD2.**
  - Is more efficient than,
  - **SORT ITAB**

- **Number of entries in an internal table** - To find out how many entries are in an internal table use **DESCRIBE**.
  - **DESCRIBE TABLE ITAB LINES_CNTLNS.**
  - Is more efficient than,
  - **LOOP AT ITAB.**
  - **CNTLNS = CNTLNS + 1.**
  - **ENDLOOP**

- **Length of a field** - To find out the length of a field, uses the string length function.
  - **FLDLEN = STRLEN (FLD).**
  - Is more efficient than
  - **IF FLD CP '*' #'.**
  - **ENDIF.**
  - **FLDLEN = SY-FDPOS.**

- **SELECT * versus selecting individual fields** - In general, use a SELECT statement specifying a list of fields instead of a SELECT * to reduce network traffic and improve performance. For tables with only a few fields the improvements may be minor, but many SAP tables contain more than 50 fields when the program needs only a few. In the latter case, the performance gains can be substantial.

  For example use:
  - **Select vbeln auart vbtyp from table vbak**
  - **Into (vbak-vbeln, vbak-auart, vbak-vbtyp)**
  - **Where...**

  Instead of using:
  - **Select * from vbak where ...**
● **Copying or appending internal tables**
  
  o Use this:
    
    - `<tab2>[] = <tab1>[]`. (*if* `<tab2>` *is empty*)
  
  o Instead of this:
    
    - *Loop at* `<tab1>`.
    - *Append* `<tab1>` *to* `<tab2>`.
    - *Endloop*

  
  ● However, if `<tab2>` is not empty and should not be overwritten, then use:
    
    o *Append lines of* `<tab1>` [from index1] [to index2] to `<tab2>`.

---

### 5.2 PERFORMANCE CHECKING

**Performance Diagnosis** - To diagnose performance problems, it is recommended to use the SAP transaction SE30, ABAP/4 Runtime Analysis. The utility allows statistical analysis of transactions and programs.
6. DICTIONARY: TABLE DEVELOPMENT STANDARDS

Whenever new tables or indexes are being created, the Coordinators must be notified and consulted. The following information should be provided to them:

- Table name
- Indexes
- Estimated number of rows
- Growth pattern
- How often accessed
- Buffered or unbuffered

Note that Z* tables should be created using data class USER or USER1 and that USER8 and USER9 classes should not be used.

6.1 TABLE NAMING CONVENTION

User Developed tables can have names that are up to 16 characters long and should follow the naming convention given below:

<table>
<thead>
<tr>
<th>Position</th>
<th>Usage</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>'Z'</td>
</tr>
<tr>
<td>2-3</td>
<td>Module/Business area</td>
</tr>
<tr>
<td>4 – 16</td>
<td>Description</td>
</tr>
</tbody>
</table>
6.2 USER TABLE DEFINITION CONVENTION

The first field defined for any LACCD developed business application table should be: **MANDT**. This is the 'client' field, and makes the table specific to the client that it is used/modified in. The only exception to this might be certain infrastructure tables that might be defined. These however, would not be business application tables.

6.3 MAINTENANCE SETTINGS

"Tab. Maint. Allowed" 'X' or (blank)

Flag if maintenance with Data Browser (Transaction SE16) is allowed.

If this flag is set, the data in the table can be changed with the Data Browser and if the user has the necessary authorization. If the data records of the table can only be maintained by program or table view maintenance (Transaction SM30), you may not set this flag.

Note: If there is a maintenance interface for the table view maintenance, the Data Browser cannot be called for the table. In this case the flag has no effect.
## 6.4 DELIVERY CLASS

<table>
<thead>
<tr>
<th>Class</th>
<th>SAP definition</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>Application Table (master and transaction data)</td>
<td>Should not be maintainable by SM30/SM31</td>
</tr>
<tr>
<td>C</td>
<td>Customizing table, maintenance only by customer, not SAP import</td>
<td>Used for Control type info, should be maintainable by SM30/SM31</td>
</tr>
<tr>
<td>L</td>
<td>Table for storing temporary data, delivered empty</td>
<td>Should not be used for LACCD tables</td>
</tr>
<tr>
<td>G</td>
<td>Customizing table, protected against SAP Update, only INS all</td>
<td>This is for use on Customizing tables developed in cooperation with SAP. SAP can INSERT entries into this type of table, but cannot change or delete any.</td>
</tr>
<tr>
<td>E</td>
<td>Control table, SAP and customer have separate key areas</td>
<td>Should not be used for LACCD tables</td>
</tr>
<tr>
<td>S</td>
<td>System table, maint. Only by SAP, change = modification</td>
<td>Should not be used for LACCD tables</td>
</tr>
<tr>
<td>W</td>
<td>System table, contents transportable via separate TR objects</td>
<td>Should not be used for LACCD tables</td>
</tr>
</tbody>
</table>

## 6.5 ACCESS CONTROL - AUTHORIZATION GROUP

This is not readily visible on the table definition screen. The definition of an Authorization Group, and assignment of one to a table, is functions done by R3-Admin. However, the identification of what that group should be is an Application Development issue. An Authorization Group has a 4 character name.
A recommendation for the name construction is 'Zxyy' where,

<table>
<thead>
<tr>
<th>Character</th>
<th>Value</th>
<th>Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Z</td>
<td></td>
</tr>
<tr>
<td>2-3</td>
<td>xx</td>
<td>2 character business area abbreviation; Example GL = General Ledger</td>
</tr>
<tr>
<td>4</td>
<td>y</td>
<td>1 alphanumeric character to distinguish subgroups within the business area for tables with common characteristics, and common users</td>
</tr>
</tbody>
</table>

Authorization group is an important attribute of a table; it is used in authorizations to grant display and update access to the appropriate SAP Users.

An authorization group should be assigned to an LACCD table if it is a delivery class 'C' (or 'G') customizing table it is an application data table (Del. Class 'A') for which access is to be restricted, even if a person has access to the application which displays / maintains it.

An Authorization group is recommended for all LACCD developed tables, to provide a mechanism for controlling access, even in cases where the table is thought to be 'public' or commonly displayable.

### 6.6 DIRECT DATABASE UPDATES OF SAP STANDARD TABLES

Under no circumstances should any program directly update SAP-delivered tables using the INSERT, UPDATE, or DELETE commands. SAP delivered tables begin with all letters other than Y and Z, and they should only be updated using an SAP transaction. To automate updates to SAP tables via a program, use the **CALL TRANSACTION** command (or optionally create a BDC using SAP supplied functions, or use bapis).
6.7 APPROVAL PROCESS FOR DIRECT UPDATE TO SAP TABLE

Very rarely, a business scenario arises for which the most viable solution is to write code that directly updates SAP table data ("Direct Update") or to make a modification to SAP source code ("Mod"). Our rule is to avoid doing either. Below is process for a developer to follow if he/she facing a situation where a Direct Update or Mod is needed:

1. Prepare the business justification as to why a Direct Update to an SAP table or Mod to SAP code is needed.
2. Identify the risks of doing the Direct Update or Mod.
3. Identify alternatives.
4. Analyze the impact of not doing the Direct Update or Mod.
5. Make the presentation to the Coordinators.
6. Depending upon the scope of the Direct Update or Mod, further approval might be required from SAP ERP Manager

In summary, when developing a table in SAP consider these questions and the answers you get for them:

- Who should be able to see the contents of this table? - Only people who have access to the transaction/report that use the table - small number of people - larger number of people - any SAP User.
- How will entries be added/changed/deleted for this table? - By an application program only - by standard table maintenance only - by both standard table maintenance and the application.
- Are entries in this table dependent on entries in another table or vice versa? Is this necessary for logical consistency? (If so, either a check table should be used to ensure this, or maintenance should only be done via an application specifically coded to ensure this relationship).
- Who should be able to add/change/delete entries of this table? - Only people who have access to the transaction/report that use the table - small number of people - larger number of people.
- Is this table one of several that all share IDENTICAL access control requirements, and will ALL be:
  - Commonly viewable by the exact same group of people?
  - Commonly maintainable by the exact same group of people?
  - The groups referred above may or may not be the same group. With this information a matrix can be established that defines - groupings of tables, types of access needed, and groups of people to have each type of access.
7. PORTAL DEVELOPMENT STANDARDS

7.1 OVERVIEW

The section of the document serves as a guideline for development and maintenance of SAP NetWeaver Portal implementations. It includes SAP Enterprise Portal Naming Standards and Procedures that must be followed during the development and maintenance cycle.

There are several reasons for maintaining and adhering to standards:

- If a standard style is used, reviews and maintenance will be more efficient.
- It shows a professional approach to development.
- It indicates quality awareness.

Abiding by these standards is required for development work to be transported to the quality assurance and production environments. Failing to adhere to it might result in the loss of development work during an upgrade.

If these standards are modified during the project, it is not required to bring the previously constructed objects up to the new standards. However, the project requires compliance with all published standards that were in effect at the time each piece of development began.

7.2 PORTAL CONTENT OBJECTS

The infrastructure of SAP NetWeaver Portal includes a number of different objects, such as roles, worksets, pages, and iViews. The objects are related, for example:

- A role can contain references to worksets, iViews, and pages as a parent object.
- A page can contain certain iViews.
- An iView can contain a reference to a system from which it reads certain data at runtime.

### 7.3 PORTAL OBJECTS NAMING STANDARDS

- For SAP imported or edited objects, do not modify name or description of migrated objects.
- Object names should be logical and follow the implementation hierarchy to make administration easier.
- Restrict the Prefix to only one i.e. aw.
- Use only lowercase letters, numbers or underscore ‘_’ for all ID’s.
- Special Characters like ‘!', '§','&','$', ‘ö', 'ä', 'ü', etc. are not allowed.
- The maximal overall length of the technical name is limited to 40 characters.
- The mandatory language is English. Please do not use words of other languages in technical names of PCD objects.

The ID of the portal component (PCD object) must be unique (Duplicate ID’s within the same folder are not allowed for an object).

Delta Links: The SAP Enterprise Portal 6.0 provides a mechanism to create links for several portal component types. A delta link is similar to a copy of an object but has a specific association with the source object (inheritance). Please use delta links to link object (e.g. iViews, Pages etc.) to Worksets and Roles. However, when utilizing an iView, Role etc. from a business package, please create a copy and paste to respective folder within your business area.

*<company name>* for all ID prefixes follows the rules below:
<object type> showed in object ID follows the naming convention below, which helps content manager identify different types of portal objects:

<table>
<thead>
<tr>
<th>Object Type</th>
<th>Name in ID</th>
</tr>
</thead>
<tbody>
<tr>
<td>Transaction (Web GUI)</td>
<td>tw</td>
</tr>
<tr>
<td>Transaction (Win GUI)</td>
<td>ts</td>
</tr>
<tr>
<td>BSP</td>
<td>B</td>
</tr>
<tr>
<td>URL</td>
<td>U</td>
</tr>
<tr>
<td>Webdynpro</td>
<td>Wd</td>
</tr>
<tr>
<td>Visual Composer</td>
<td>Vc</td>
</tr>
<tr>
<td>Par File</td>
<td>P</td>
</tr>
</tbody>
</table>

The System name, ID or alias may contain up to 20 characters of any type, including spaces, but not apostrophes and quotation marks. Use lower case.

For extension of any collection of predefined terms contact the Portal Governance Team.
7.3.1 ID PREFIX FORMAT FOR PORTAL CONTENT OBJECTS

1. A complete ID for a role without a namespace prefix:
   
   \textit{pcd:portal\_content/administrator/content\_admin/content\_admin\_role}

2. A complete ID for a role with a namespace prefix:
   
   \textit{pcd:portal\_content/com.sap.pct/administrator/content\_admin/com.sap.portal.content\_admin\_role}

3. domain.company.object type.source.type

   \textit{edu.laccd.pct.benefits.role}

Rules:

- The length is limited to 100 characters.
- The dot (.) is the separator
- Use only small letters, numbers and underline
- The registration of all portal components is mandatory

7.3.1.1 PROJECT NAMESPACES

The general namespace used for SAP Enterprise Portal is:

- \textit{domainidentifier.companyID.pct.projectID}

The domain identifier and company ID identify the object as belonging to a specific company namespace.

For LACCD,

- domain identifier is \textit{edu},
- companyID is \textit{laccd}
- pct abbreviation is used to indicate that an object is portal content
- projectID is used to identify objects that belong to a single project.
All objects or content elements that are contained in a business package share the namespace prefix of this project:

domainidentifier.companyID.pct.projectID.objectID

**Example:** com.yourcompany.pct.lists.myevents

All objects developed in LACCD should share the namespace prefix as below

edu.laccd.pct.<projectID>.objectID

Given below is the namespace prefix convention to be followed for various portal content objects.

### 7.3.1.2 ROLES

**Name:** <Description> (All CAPS, some CAPS, Lower Case, any combination)

E.g. Compensation Planning

**ID Prefix:** <domain>.<company>.<object type>.<source>.<type> (all lower case)

E.g. Edu.laccd.pct.benefits.role

**ID:** <object type>_<object description> (all lower case)

E.g. Compensation_planning

### 7.3.1.3 PAGES – IVIEWS

**Name:** <Description> (All CAPS, some CAPS, Lower Case, any combination)

E.g. Time Entry

**ID Prefix:** <domain>.<company>.<object type>.<source>.<type> (all lower case)

E.g. Edu.laccd.pct.benefits.page

**ID:** <object type>_<object description> (all lower case)

E.g. B_time_entry
7.3.1.4 WORKSETS

**Name:** Description (All CAPS, some CAPS, Lower Case, Any Combination)

E.g. Skills Profile

**ID Prefix:** <domain>.<company>.<object type>.<source>.<type> (all lower case)

E.g. Edu.laccd.pct.benefits.workset

**ID:** object description (all lower case)

E.g. Skills_profile

7.3.1.5 FOLDERS

**Name:** Description (All CAPS, some CAPS, Lower Case, Any Combination)

E.g. Employee Self Service

ID Prefix: empty

**ID:** object description (all lower case)

E.g. ESS

7.3.1.6 SYSTEMS

**Name:** nameofsystem_Type_SID_clientid (All CAPS, some CAPS, Lower Case, any combination)

E.g. HRD_DEV_RD1_100

BW_SBX_BD1_120

**Prefix:** <domain>.<company>.<object type>.<source>.<type> (all lower case)

E.g. edu.laccd.pct.benefits.system

**ID:** nameofsystem_type_clientid (all lower case)

E.g. hrd_dev_rd1_100

**Alias:** nameofsystem_Type_SID_clientnumber (All CAPS)

E.g. HCM_DEV_EC0_100

BW_SBX_BI0_010
7.4 TRANSPORTS

Create a folder to house the transport package. For example if you are creating a transport package for ESS then the folder name would be ESS_062105 <create date>.

Navigate to the folder created above and right click to create the transport package. Name the transport package as ESS_062105_01 (02, 03 for subsequent transports for the same day).

This way all transports will have a sequence when building production by date stamp (as created) and time stamp as saved on the OS level.

7.5 JCO DESTINATIONS

Name: Name of the jco Connection (All CAPS, some CAPS, Lower Case, Any Combination)

E.g. LACCD_ERP_humanresources_metadata (for Meta data)
LACCD_ERP_humanresources (for model data connection)

7.6 WEBDYNPRO FOR JAVA NAMING STANDARDS

7.6.1 DEVELOPMENT ENTITIES

{a} - Application
{act} - Action
{c} - Component
{cc} - Custom controller
{dc} - Development Component
{m} - Model
7. Portal Development Standards

7.6.2 CONTEXT ENTITIES

- Context attribute
- Context child node
- Context node
- Context name; always the same name as the controller to which it belongs
- Model node
- Model object (referred to by a model node)
- Recursive node
- Value attribute
- Calculated value attribute
- Value node

Abbreviations for generic and composite entities

- Data type defined either in standard Java or by the webdynpro data dictionary
**Abbreviations for subscripts of composite entities using the suffixes recommended by SAP**

<table>
<thead>
<tr>
<th>Subscript</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>{na}</td>
<td>Application = {a}App</td>
</tr>
<tr>
<td>{nc}</td>
<td>Component controller = {c}Comp</td>
</tr>
<tr>
<td>{nccc}</td>
<td>Custom controller = {cc} Cust</td>
</tr>
<tr>
<td>{nciv}</td>
<td>Component interface view = {w}interfaceview</td>
</tr>
<tr>
<td>{nctl}</td>
<td>Controller of any type</td>
</tr>
<tr>
<td>{nm}</td>
<td>Model = {m}Model</td>
</tr>
<tr>
<td>{npi}</td>
<td>Inbound plug = {pi}In</td>
</tr>
<tr>
<td>{npo}</td>
<td>Outbound plug = {po}Out</td>
</tr>
<tr>
<td>{npr}</td>
<td>Project = {pr}</td>
</tr>
<tr>
<td>{nsi}</td>
<td>Standalone component interface = {si}compi</td>
</tr>
<tr>
<td>{nsiv}</td>
<td>Standalone component interface view = {si}{siv}</td>
</tr>
<tr>
<td>{nu}</td>
<td>Component usage = {nc}{p}Inst</td>
</tr>
</tbody>
</table>
The look and feel of portal should be designed in the way to reflect corporate identity, enhance user experience, promote usability of the portal by improving the process by which a user gets a task accomplished including providing a platform for knowledge sharing and collaboration.

A good portal design will increase the usability of the portal provide a successful framework for future growth. As a portal content manager or developer, it is critical to follow the design standards and guidelines to develop the user interface, navigation, personalization features and collaboration features on the portal.

**User Interface**

The general guidelines for graphic user interface design should ensure the user has the best possible orientation within the application and application is as consistent and harmonious as possible.

**Layout Techniques**

<table>
<thead>
<tr>
<th>Personalization</th>
<th>Movable Pieces</th>
<th>Allow users to rearrange their portal content</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>Within SAP EP, you can set various levels of personalization properties for the iViews and Pages</td>
</tr>
<tr>
<td></td>
<td></td>
<td>As a best practice, avoid deploying/implementing personalization to your users the first time you go live to give the users</td>
</tr>
</tbody>
</table>
### 7. Portal Development Standards

#### SAP Development and Quality Assurance Policies and Procedure Document

<table>
<thead>
<tr>
<th>Personalization Link</th>
<th><strong>In EP6.0 there is an Add to Favorites link on the portal that enables the user to save the page, iView they are currently navigating as a favorite so that they can login to the same page/iView at a later time instead of navigating to the same after logging in. The users should be trained to use this feature.</strong></th>
</tr>
</thead>
</table>

| **Portal Pages** | **Visual Framework** | **Consistency in colour, fonts and writing style – Develop the portal theme that is consistent with the rest of organization branding and color schemes. Build the portal to cater to all types of users and all types of screen resolutions by adjusting the font sizes etc.** |
|-------------------|----------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------
| **Component Layout** | **Arrange iViews on a page in an organized fashion and layout that is easy to navigate and appealing to the eye.**  
|                      | **Use diagonal balance when you have a title or header at the top, and some links or action buttons such as OK and Cancel, or Submit, or Back and Next at the bottom.**  
| **UI Controls** | **Closable Panels is another space-saving device that depends on a user’s click. Within EP, enable maximize and minimize properties for iViews and Pages.**  
|                      | **This enables you to provide a Dynamic UI as the user follows each**  

### Page Layout

- The sequence of page elements should be organized in a vertical or horizontal fashion, which takes care of:
  - Flow of control
  - Dependencies
  - Information about which elements belong together
- Nesting of page elements should be arranged in a hierarchical or top-down fashion ensures:
  - Dependencies
  - Togetherness
- At different hierarchy levels spacing of page elements establishes esthetics and proper application of how elements should be perceived.

### Flow of control

The focus of activity should move across a screen or page while the user performs a certain task. The flow control is important in two respects:

- Efficiency in performing a task
  - The transparency and understandability of a screen or page

### Presentation

<table>
<thead>
<tr>
<th>Center Stage</th>
<th>Information centered or task centered</th>
</tr>
</thead>
<tbody>
<tr>
<td>Titled Sections</td>
<td>Group content according to similarity and then choose a presentation layer</td>
</tr>
</tbody>
</table>
The following left diagram is an example of good design of control flow, in which the reading direction is from left to right and from top to bottom.

![Flow of Control Diagram]

**Figure 1: Flow of Control**

**Portal Pages**

Page Layouts are used within EP to structure the page content and enable you to organize content in the most effective way:

- Information displayed should not require the user to scroll horizontally and all efforts should be made to prevent the user from having to scroll vertically.
- The number of iViews on a page should be limited to five.
- In the case of pages with URL iViews, the page should be single column with the single URL iView assigned to the page (full page layout only).

A portal page holds iViews. Before creating a page, the following issues need to be considered.

- Each iView has an optimal display mode, depending upon the information it contains.
- The page might be assigned as a dynamic navigation iView to another page, which can appear by default in the portal navigation panel.
• Before assigning iView to a page, consider one of the following issues that affect page performance:
  
  o The cache level of each iView, which determines from where iView data is retrieved.
  
  o The isolation method of each iView, which refers to the relationship between an iView and its page, and iView behavior on the page. The isolation method determines whether iView content is collected at the server or at the client, and how the PageBuilder component displays the content. Either URL or Embedded isolation mode is suitable for iViews with relatively heavy content. Embedded isolation mode is for:

  ▪ iViews that are tightly coupled (with POM events) and need to refresh themselves, through the server, according to mutual client actions
  
  ▪ iViews that are presented as navigation nodes outside a page
  
  ▪ Regular iViews that will not be added to a Web Dynpro page

URL isolation mode is used for:

  ▪ When presenting content external to the portal server
  
  ▪ For regular iViews that will be assigned to a Web Dynpro page

**iView Defaults**

• iViews should be displayed in a maximized view by default.

• Users should not have to click on an iView to display the corresponding data; the iViews should be open and available when the user accesses the page containing the iViews.

• iViews should be sized automatically except for in the case where iViews display text only, for which the iView sizing can be pre-set.
- URL iViews sizing should be set to full page.

**Spawning New Browser Windows**

As a practice, iViews should not be launched in new browser windows. If there is a business requirement to launch an iView in a new browser windows, approval must be obtained from the business owner of portal and iView height/width should be adjusted to provide the user with a viewable/usable application.

**Charts & Graphics**

Please refer to [http://www.sapdesignguild.org/resources/diagram_guidelines/index.html](http://www.sapdesignguild.org/resources/diagram_guidelines/index.html) for guidelines for the use of charts, graphics, images, colors and text.

**Accessibility**

The portal interface should be tested ahead of time to make sure all the users can access the portal (login page) and the content within the portal. There are specific tools available for testing the portal interface like httpwatch to troubleshoot accessibility issues by understanding where the browser requests are being routed to from the portal and what backend systems are being accessed.

**Personalization**

Personalization refers to those areas in the portal where users are allowed to change the settings to their preferences. As a best practice, you should not allow users to personalize their portal in the first phase or first go-live to enable the users to learn the new UI and get used to portal navigation. Once users are comfortable, personalization features can be added. On an iView or a Page the following features can be allowed for user to personalize the portal:

- Personalization
- Remove object
**Portal Themes**

A portal administrator can maintain and create multiple themes or branding on the portal. As a best practice the display rules must be set for users/groups of users so that they get a particular theme by default. Portal personalization features for changing themes by users should be turned off.

**Portal Languages**

For initial roll-outs, work with a single language – English for example. Rolling out a multi-language portal requires additional work and tuning the portal content to different languages as well. Approval must be obtained from the competency center before evaluating a multi-language portal.

**User Mapping**

As a best practice a single-sign on solution must be deployed to the users seamlessly. In certain situations it may be required that users maintain user mapping on the portal. This portal personalization feature must be enabled on the portal and users should be allowed to maintain their user mappings based on content they are assigned to. Users will be responsible for maintaining their user id and password information in sync with the associated back-end system password.

However, maintenance of user mapping by users can be a difficult task, as users will have to change their mapped information if they were to change the password in the backend system, in such cases, user mapping via trusted relationship in a reference system must be used.

**Work Protect Mode**

Work Protect Mode is used on the portal to save/cache the session the user is in and release the same when the user navigates away from the page. The default setting for Work Protect Mode should be used and users should not be allowed to change this setting.
Navigation

SAP Portal provides a role-based user interface and the ability for administrators to control the navigation layout. As a best practice you should avoid overloading the users’ with a lot of roles and maximize their navigation experience by providing the users’ with the information they need to perform their tasks/jobs.

Navigation Area

SAP portal offers a comprehensive navigation environment for retrieving the information users need to perform business functions. By default SAP Portal consists of three levels of navigation – Top Level Navigation (associated with top level roles), Second Level Navigation (associated with worksets) and Detail Navigation (associated with pages and iViews).

The picture below illustrates the navigation areas of the Enterprise Portal.

![Figure 2: Portal Navigation](image-url)
Number of Clicks

The general rule of thumb is a user should not have to drill down more than four navigation levels to reach the information they desire. This is not a rigid rule, in some cases five clicks are permissible; beyond five mouse clicks should be avoided. The goal of this approach is to simplify navigation and minimize any frustrations user may have when working in the enterprise portal. A good portal design dictates to avoid multiple levels of detail navigation. In order to better organize content on the portal, you may want to look at options that enable you to make your worksets as entry point on the portal (first level navigation) instead of the role, again trying to maximize user interaction and satisfaction on the portal.

Top Level Navigation Organization

As shown in the figure above the top level navigation area represents the role entry point. All roles assigned to the user and defined as entry points are displayed in this area.

- No more than eight roles tabs are recommended to display for any user with the exception of test users.
- The number of role tabs (first level navigation) displayed should not require the user to scroll horizontally to display all tabs.
- The role entry points are defined by the project teams with approval of portal owners.

The second level of the top level navigation displays worksets. Worksets represent a grouping of pages and iViews into a logical workgroup.

- The standard for workset tab displayed in the second level of top navigation is recommended at no more than eight (8) worksets.
- The number of workset tabs (second level navigation) displayed should not require the user to scroll horizontally to display all tabs.

The detailed navigation area displays pages, iViews and nested worksets.
Usability

The key issue why some portal implementations fail is because they are not usable, functional or do not provide relevant content/information to the users based on their roles. As a best practice, spend time on usability. Work with your users to make the portal relevant to their jobs/roles in the organization. Put in a process to incorporate user feedback into the portal development process. You can do this by creating surveys, questionnaires etc.

General Portal Help

General portal help is accessed from the help link in the upper right of the Welcome Banner area of the portal. The default link for help is http://help.sap.com; and you should modify this link to point to internal custom help files or a help site.

Application Specific Help

Application specific help can be defined for each role, iView/Page. Context Sensitive Help can be designed based on user role within the organization. As a best practice avoid implementing iView/Page level help to minimize impact on the support organization.

Object Navigation Options

The object navigation options button is located at the upper right of the active iView/Page tray. The standard options available can be customized. A portal administrator can set these options for each object (iView/Page) on the portal as noted below:

- Detail
- Open in New Window
- Personalize
- Refresh
- Remove
As a best practice, the personalize and remove options should not be activated the first time your go-live. The detail help option should be de-activated as well as it provides details on iView/Page location that may not be relevant to the user.

**Portal Back and Forward Buttons**

Portal Back and Forward links are used to move back and forward through the portal page history, not the browser history. Based on requirement these can be set to display/non-display. As a best practice these options should be provided to the users as the browser back/forward buttons are not synchronized with portal applications in general. The user should use the portal/application specific back/forward links/buttons to navigate through transactions.

**Search Functions**

TREX search capability can be integrated on the portal with the search box in the navigation area at the top. TREX indexes can be deployed to users through role assignments on the portal. It is recommended that separate TREX servers/engines are used for external and internal access. Files/Content Repositories available to external users can be located on a file server residing in the DMZ. These files should be indexed by a TREX server located in the DMZ. External users will have access only to the TREX indexes located in the TREX server in the DMZ. Files/Content Repositories located on file servers behind the firewall should be indexed by a TREX server behind the firewall. These indexes will be available only to internal employees in addition to the external indexes.

**Collaboration**

**Collaborative Processes**

The design of portal collaboration largely depends on the processes of collaboration - processes of communication, coordination, cooperation, but also information sharing. These processes do not work independently of one another but are usually intermingled and determined by each other. The following table illustrates collaborative techniques differentiated by the collaboration process they support.
<table>
<thead>
<tr>
<th>Communication</th>
<th>Coordination</th>
<th>Cooperation</th>
<th>Information Sharing</th>
</tr>
</thead>
<tbody>
<tr>
<td>• E-mail</td>
<td>• Workflow management</td>
<td>• Electronic meeting systems</td>
<td>• Whiteboards</td>
</tr>
<tr>
<td>• Audio- and videoconferencing</td>
<td>• Calendar and scheduling</td>
<td>• Group authoring software</td>
<td>• Application sharing</td>
</tr>
<tr>
<td>• Telephone</td>
<td>• Project management</td>
<td></td>
<td>• Knowledge management</td>
</tr>
<tr>
<td>• Instant messaging</td>
<td></td>
<td></td>
<td>• Threaded discussions</td>
</tr>
<tr>
<td>• Chat</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Fax</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Screen sharing systems</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Collaboration Rooms**

Collaboration Rooms are based on templates. The structure and the content of the Room are predefined, which makes the creation process very convenient for the portal user. The template concept enables a company to define best practices and standard processes for project management, thus supporting their work groups with "off-the-shelf" Collaboration Rooms.

Users who have authorization for the Collaboration Room have a Rooms entry point in the top-level navigation. In the second level navigation, users have an overview page where they can access the Rooms of which they are members. Additionally, the user can display their favorite Rooms in the second level navigation for faster access.

Navigation between individual Collaboration Rooms is based on standard navigation on the portal. Each Room consists of a workset, which contains pages and iViews. The pages of a
workset can be structured in a flat list or, if the room is a large one with lots of pages/documents/information, in a hierarchical structure. The detailed navigation allows users to move from page to page, as they are used to in the standard portal.

**Development Tools**

This section should give an overview of which development tools or systems will be used to facilitate the differing aspects of the development processes. This section should include the toolsets to be utilized for development based on different scenarios, e.g. when to use Visual Composer.

SAP provides a number of tools and resources to assist developing applications on the SAP NetWeaver platform. Some of the tools and resources available to developers are:

- SAP NetWeaver Developer Studio (NWDS)
- SAP Visual Composer
- SAP Enterprise Portal Developer Kit (PDK)

NWDS provides additional tool and perspectives, such as Java/J2ee, Web Dynpro and Web Services etc.
8. BUSINESS WAREHOUSE

When coding the report program, the Developer should use the corresponding module’s standard report header. You can use a standard report header by calling function module ‘Z_REPORT_HEADER_LACCD’. Programs should include a listing of the selection criteria used to generate the report (optional).

In order to improve performance in terms of decreasing the extraction time and data volumes, data extraction from R/3 should be delta extraction (Delta updates) as compared to an extraction of the entire dataset (Full update) under most circumstances.

8.1 BW OBJECTS NAMING STANDARDS

The purpose of defining strict naming standards for BW objects is to ensure the entire project team is consistent in the approach to creating and identifying objects in the BW system. The following BW objects are covered by this document.

<table>
<thead>
<tr>
<th>Subject</th>
<th>Definition</th>
<th>Naming</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>Infocube</td>
<td>An infocube is the central data container that forms the basis for reports and analyses in BW. Infocubes contain two types of data: key figures and characteristics.  An infocube is a set of relational tables that are arranged in a star schema with a large fact table for recording transaction data at the center and several dimension tables around the</td>
<td>Zff_Cnn, where ff = functional area, nn = two-digit number of cubes</td>
<td>ZSD_M50 where, Z - is the constant SD - functional area of cubes that feed the multi-cube M - multi cube, and 50 is the number of the cube</td>
</tr>
<tr>
<td><strong>Fact Table</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>----------------</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>The fact table contains the key figures of the infocube while the dimension tables contain the characteristics of the cube. Infosources (see below) supply data to infocubes.</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th><strong>Infosource</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>An infosource is a set of logically associated information which can contain transaction data (stored in infocubes) and master data (attributes, texts, and hierarchies stored in separate tables).</td>
</tr>
<tr>
<td>Infosources describe all the information available for a business transaction or type of business transaction (for example, cost center accounting).</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th><strong>Transaction Data:</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>Infosource = datasource</td>
</tr>
<tr>
<td>Technical Name Long Description = Datasource Description</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th><strong>Master Data:</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>Select infoobject, the technical name and description</td>
</tr>
<tr>
<td>Will be assigned from the infoobject.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th><strong>ODS</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>An ODS object contains supporting information for the BW infocubes. It may be used to contain information at a more detail level than the summarized infocube information; or it may contain information combined from multiple sources. This data is accessible via queries and the Bex analyzer and browser, or via infoset query.</td>
</tr>
<tr>
<td>Zff_Onna</td>
</tr>
<tr>
<td>Ff = functional area</td>
</tr>
<tr>
<td>Nn = two-digit number</td>
</tr>
<tr>
<td>A = A, B, C, D, etc for multiple ODS's that feed the same infocube</td>
</tr>
<tr>
<td>If multiple ODS objects were created to support ZCCA_002. They would be called ZCCA_02A and ZCCA_02B. (because of 8 character limitation on technical name.)</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th><strong>Infoobject</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>An infoobject is a generic term for characteristics and key figures in the infocube.</td>
</tr>
<tr>
<td>Custom infoobjects should always start with a Z.</td>
</tr>
<tr>
<td>A copy of the 0MATERIAL infoobject</td>
</tr>
<tr>
<td>Business Information Warehouse. Infoobjects are used in infocubes and in the three structures that are relevant for data requests—extract, transfer, and communication structures.</td>
</tr>
<tr>
<td>---</td>
</tr>
<tr>
<td>Query</td>
</tr>
<tr>
<td>A custom query for infocube ZSD_C04 has the technical name QZSD_C04_5000</td>
</tr>
<tr>
<td>Query View</td>
</tr>
<tr>
<td>Web Template</td>
</tr>
</tbody>
</table>
### Roles
A role in BW identifies a person responsible for a specific business area.

**Role Type (S-Single, C-Composite)**
- **Ff** = functional area (SD, FI, etc.,)
- **dd** = brief description of role

---

### Variables
Variables are parameters of a query that are set in the query definition and are not filled with values (processed) until the query is executed and inserted into a workbook.

They function as a store for characteristic values, hierarchies, hierarchy nodes, texts and formula elements and can be processed in different ways.

Variables serve for the flexible setting of queries.

**Variables**
- **Y_nnnnn**
  - **S** = Selection option variable (range with include/exclude/insert)
  - **I** = Interval variable, i.e. The user enters a range of entries
  - **M** = Multiple single values
  - **P** = Parameter variable (single value)
  - **V** = Pre-calculated value set variable
  - **T** = Text variable
  - **F** = Formula variable
  - **H** = Hierarchy variable
  - **N** = Hierarchy node variable
  - **Nnnnn**: Meaningful name based on the infoobject for which the variable is used (max of 5 characters)

---

and add _nn, (nn = two-digit sequential number).
| Infopackages | Infopackages are the method that BW uses for loading data from a source system into BW. They are associated with an infosource and a source system. They are used to load either transactional or master data. They can be combined into infopackage groups. | Infosource_tttt_X  
Infosource = 0MATERIAL  
Tttt – Type of data (TRANS – transaction, TEXT-Text, ATTR-Attribute, 01-Heirarchy (02, 03 for multiple hierarchies)  
X for Type of Update  
I = Delta Initialization.  
F = Full Update.  
D = Delta | To load employee attributes from HR for the proof of concept, the infopackage would be:  
0EMPLOYEE_ATTR HR F |
| Flat Files | Used to copy files from external sources into BW. | Infoobjectname.csv  
Cdpindicator.csv |
| New Extract Structures | Used to initially hold data when extracting from the source system | IO XXXX  
IO = infoobject  
XXX = TEXT, HIER, ATTR |
| New Transaction Datasource | Used to provide transaction data from R/3 for delivery to BW. | IO XXXX  
IO = datasource  
XXX = TEXT, HIER, ATTR |
| Aggregates | Used to pre-summarize data to improve data reporting performance | |
| Info Object Catalog | Used to group characteristics & Key Figures | Zinfocubetechname XXX99  
Infocubetechname = Technical name of the infocube  
XXX = CHA for characteristic or KYF for key figure |

8. Business Warehouse

LACCD| Confidential Version 2.0 - 2013
<table>
<thead>
<tr>
<th>Infoprovder</th>
<th>Zmmff_xnnMm = primary module (FI, HR or ST) Ff = functional area (minus hyphens, i.e. CO-PA use CO) X = type of infoprovder, M = multiprovider C = infocube R = remotecube V = virtualcube I = infoset. Nn = two-digit number</th>
<th>ZFICAC02</th>
</tr>
</thead>
<tbody>
<tr>
<td>Datastore</td>
<td>In BI 7.0 ODS (Operational Data Store) object has been renamed to DSO (Data store Objects). There are three types of this new object: a. Standard: similar to the old ODS, has three tables for activation queue, table for active data and change log table. b. Write optimized: has only active data table (with delta capture capability) c. Direct update: consists of active data table only. SIDS are not generated for write optimized and direct update DSO's.</td>
<td>Zffffff_Onna Mm = primary module (FI, HR or ST) Ff = functional area (minus hyphens, i.e. CO-PA use CO) Nn = two-digit number A = A, B, C, D, etc for multiple datastore object’s that feed the same infocube. A custom datastore object has a limit of 8 characters for the technical name. If a limitation occurs when naming the datastore object please try the following: Eliminate the 0 after O, i.e. ZFIPUO1A</td>
</tr>
</tbody>
</table>
| Structure | Scube_nnnn  
Cube = infocube technical name  
Nnnn = sequential number starting at 5000 | A structure that is used in the Sales and Distribution infoprovider  
0SD_o01 would have the technical name  
S0SD_001_5000 |
|----------|-------------------------------------------------|-------------------------------------------------|
| Report   | Super User reports:  
Zcube_Rnnnn where  
Cube = infoprovider Name  
R = constant for reports  
Nnnn = four-digit sequential number (range 5000 to 9999)  

Power User reports:  
Ycube_Rnnnn_DDD_III,  
where  
Cube = infoprovider Name  
R = constant for reports  
Nnnn = four-digit sequential number (range 5000 to 9999)  
DDD = Department abbreviation  
III = username of power user | |
| Restricted Key Figure | Rkcube_nnnn  
Cube = infoprovider technical name  
Nnn = sequential number starting at 5000 | Custom restricted key figure for infoprovider  
0FIFM_C01 would have the technical name  
RK0FIFM_C01_5000 |
<table>
<thead>
<tr>
<th>Calculated Key Figure</th>
<th>Ckcube_nnnn</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Cube = infoprovider technical name</td>
</tr>
<tr>
<td></td>
<td>Nnnn = sequential number start at 5000</td>
</tr>
<tr>
<td></td>
<td>Custom calculated key figure for infoprovider 0FIFM_C01 would have the technical name CK0FIFM_C01_5000</td>
</tr>
<tr>
<td>Process Chain</td>
<td>Z_dddddddddddddd</td>
</tr>
<tr>
<td></td>
<td>Ddddddddddddddddddd</td>
</tr>
<tr>
<td></td>
<td>dd = description of process</td>
</tr>
<tr>
<td></td>
<td>: Z_LD FICO MD D – Delta load of master data for FI</td>
</tr>
<tr>
<td>Application Component</td>
<td>Application components should be named Z*. There is a textual description that should explicitly provide information on the function</td>
</tr>
<tr>
<td></td>
<td>ZPOCAREA</td>
</tr>
<tr>
<td>Custom Generic Extractor</td>
<td>Z_xx_(ffff + description if transactional),(infoobject if master data)Xx for type of extractorTr = transactional data.M = master data attributes.Tx = master data TextFfff = functional area or infoobject</td>
</tr>
<tr>
<td></td>
<td>Z_M_ZPLANT</td>
</tr>
<tr>
<td>Datasource</td>
<td>Zdatasource_xxxx, where xxxx = TRAN, TEXT, HIER, ATTR</td>
</tr>
<tr>
<td>DTP Naming</td>
<td>Data transfer process (DTP) is used to transfer data within BI from one persistent object to another object, in accordance with certain transformations and filters. We use the</td>
</tr>
<tr>
<td></td>
<td>For naming of DTP following conventions should be followed – a. For DTP with filters defined in it, we give a</td>
</tr>
<tr>
<td></td>
<td>Example : With filter - TERM 20042 – 20061 Without filter - Position history</td>
</tr>
<tr>
<td></td>
<td>Z_M_ZPLANT</td>
</tr>
<tr>
<td>Data transfer process (DTP) to transfer data from source objects to target objects in BI. We can also use the data transfer process to access infoprovider data directly.</td>
<td>Very short description followed by period load</td>
</tr>
<tr>
<td>b. For DTP without filters, just give a short description</td>
<td></td>
</tr>
</tbody>
</table>

## 8.2 BROKEN REPORTS

If a BW report is broken then it needs to be taken out of service from Portal. In such cases, the links and the description of these reports should be removed from the portal page by the Portal Team immediately.

The reports description is maintained by using the tcode ZBW_TABLE in BW system.

The changes should be made in the DBW system and then need to be moved to Production through the Transport landscape.

Once the report is fixed, it should be tested in the DBW system, and the report link its description has to be restored on the portal page.
9. LACCD STANDARD EMAIL TEMPLATE

The following LACCD email standard template should be used in all the programs developed in future:

<table>
<thead>
<tr>
<th>From Address</th>
<th><a href="mailto:XX_YYY_no_reply@laccd.edu">XX_YYY_no_reply@laccd.edu</a></th>
<th>XX represent SAP modules</th>
</tr>
</thead>
<tbody>
<tr>
<td>YYY</td>
<td>optional for e.g CATS</td>
<td></td>
</tr>
</tbody>
</table>

Restrict receiver's reply to the “From Address”

<table>
<thead>
<tr>
<th>Subject line</th>
<th>LACCD – xxxxxxxxxxx (50 characters)</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>Salutation</th>
<th>Dear Full name, (IT1-name)</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>Body information</th>
<th>First line</th>
<th>This message is provided by the Los Angeles Community College District.</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>Access Method (optional)</th>
<th>SAP access through portal -</th>
</tr>
</thead>
<tbody>
<tr>
<td>E.g. (For SAP Portal Access (for those who prefer or have access to the electronic Timesheet via the Portal) click here <a href="https://portal.laccd.edu/irj/portal">https://portal.laccd.edu/irj/portal</a> to access through portal. From the Home page click My Time Sheet Entry under Employee Services or go to Employee Services tab -&gt; working time -&gt; My Time Sheet Entry and Press Enter Times or F5 function key.</td>
<td></td>
</tr>
</tbody>
</table>

<p>| SAP access through attachment - |
| E.g. (For SAP GUI Access (for those who prefer or only have the GUI) Double click the attachment above to access your electronic timesheet through SAP GUI.) |</p>
<table>
<thead>
<tr>
<th>End note</th>
<th>This mailbox does not have an attendant. Do not reply to this message.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Contact information (optional)</td>
<td>E.g Contact your Manager or TimeKeeper to enter time for those period</td>
</tr>
<tr>
<td>Initials (Finishing)</td>
<td>Department name / Individual person</td>
</tr>
<tr>
<td>Confidentiality (optional)</td>
<td></td>
</tr>
<tr>
<td>Last line (leave 3 line space)</td>
<td>This email is created and powered by the LACCD SAP information technology team.</td>
</tr>
<tr>
<td>Attachment Name</td>
<td>Meaningful attachment name</td>
</tr>
<tr>
<td>Attachment Name</td>
<td>e.g SAP log on/PO display/Timesheet display/Time approval etc</td>
</tr>
</tbody>
</table>

Note: Text in blue color should not be changed.
An example of a standard email to be followed is given below:

From: TM_CATS_noreply@laccd.edu
Subject: LACCD-TimeSheet missing entry

Dear Mr. Harlan Darius Penn,

This message is provided by Los Angeles Community College District.

You are missing time entries for the dates below. Please revisit your timesheet and complete it.

<table>
<thead>
<tr>
<th>Date</th>
<th>Actual Hours</th>
<th>Time Entered</th>
</tr>
</thead>
<tbody>
<tr>
<td>07/07/2010</td>
<td>8.00</td>
<td>0.00</td>
</tr>
<tr>
<td>07/02/2010</td>
<td>8.00</td>
<td>0.00</td>
</tr>
<tr>
<td>07/01/2010</td>
<td>8.00</td>
<td>0.00</td>
</tr>
<tr>
<td>06/30/2010</td>
<td>8.00</td>
<td>0.00</td>
</tr>
<tr>
<td>06/29/2010</td>
<td>8.00</td>
<td>0.00</td>
</tr>
<tr>
<td>06/28/2010</td>
<td>8.00</td>
<td>0.00</td>
</tr>
</tbody>
</table>

To access the electronic Timesheet:

For SAP GUI Access (for those who prefer or only have the GUI), double click the attachment above to access your electronic timesheet through SAP GUI.

For SAP Portal Access (for those who prefer or have access to the electronic Timesheet via the Portal) click here https://portal.laccd.edu/irj/portal to access through portal. From the Home page click My Time Sheet Entry under Employee Services or go to Employee Services tab -> working time -> My Time Sheet Entry and Press Enter Times or F5 function key.

Please make a note that you cannot enter time more than 4 weeks prior to current date. Contact your Manager or Timekeeper to enter time for those period or any other questions.

This mail box does not have an attendant. Do not reply to this message.

Thank You,

Time Department
10. TRANSPORTS STANDARDS

10.1 OVERVIEW OF LACCD TRANSPORT PROCESS

All Transport request must be approved through HQ Quality Center. The Transport_Requests project is in the QC_Management domain.

The transport process is as below:

1. Once the SAP Team members have configured their changes in the appropriate DEV client and the change is released for transport, they should fill out the Quality Center Transport Request form for approval from the First Approvers. Transports to QAS, QBW, QEP can be initiated at any time by the team member. QA transports will need approval from the Team Lead or Manager listed under ‘First Approvers’ field to move the transport to QAS.

Prior to requesting a transport to QAS, the responsible tester should complete the following steps in HP-QC ‘Issue_Management’ project:

- Create a test case for the issue, and complete the unit testing.
- Link the test case to the issue from the Test Plan.
- Execute the test case and update the results in Test Lab module in QC.

2. The Team Lead or the Manager will receive a notification request via email for Transport Approval. The team lead or the Manager will then verify the following before approving or declining the request:

- The change was correctly made to the system and tested
- That other integrated areas are not affected.
- That the transport is released in SAP.
Also, verify the following information on the Transport_Request form:
- Transport Client Dependency
- Source client
- Target clients
- Issue Log #
- Transport Description, and
- Urgency

3. After verifying all the information, the First Approver will approve the request to be transported from the Dev server to the QAS server. However, if adequate information is not provided, the approver can ‘Decline’ the transport request. The transport will have to be resubmitted by the team member.

4. If the request is approved to move to QAS, the Basis Team will receive a notification to move the Transport from DEV to QAS.

5. The Basis Administrators will perform the transport, update the results in the Transport Request form, whether the transport was “Successful”, had a “Warning” or had a “Fatal Error”. Notification is then sent to the SAP team member, the functional lead and project manager about the status of the transport execution.

6. Transport request to PRD without testing in QA server will not be approved.

Once the unit testing is complete and the test case is passed in DEV, the SAP team member should select a QAS tester, and the assigned tester then tests in QAS and updates the Testing Status Field to ‘Complete’ in Quality Center indicating they have tested the transport.

7. The SAP team member will then request for PRD approval by selecting the destination and the ‘Final Approver’.
8. The above steps must be followed for transports from QAS to PRD. The transport to PRD will have to be approved by the Final Approver.

9. A Team Lead can approve all transports related to configuration or ABAP programs that impact their functional domain.

10. BW transports are approved by the BW -zCoordinator for QAS and by the Manager for PRD. This includes BW objects, extractors and programs.

10. Security related transports are approved by the Functional Lead or Functional Manager.
10.2. TRANSPORT SCHEDULE

There is an established daily schedule for moving transports during the weekdays. Special schedules for short days and holidays are established by the Application team manager as and when needed. Depending on the urgency, transports can also be moved in the next 30 minutes, or 2 hours or anytime.

The regular End of Day transports takes place at 4:00 PM on every weekday.

Emergency transports will be viewed as an exception to the regular ‘end of the day’ transport schedule and moved in a timely manner.

In order to request an emergency transport, the initiator selects “Emergency Transport” in the Urgency field of the transport request form. This request maybe be accompanied by a call to the basis administrators from the Approver or SAP Team member.

A Transport routing process is depicted in the chart below:
LACCD SAP Transport Routing

10. Transports Standards

LACCD| Confidential Version 2.0 - 2013
10.3 TRANSPORTS DESCRIPTION FORMAT

The Transports Description field is 60 characters in length. It is recommended to use the following format for naming transports:

<Module>-<QC Issue Log #>-<Track>-<Description>

Eg: HR-3632-IT0016 copied on separation action-01-Remove IT0016

Where,

Module: SAP Module (FI, HR, MM, SD, etc.)

Track: Sequence of transport

10.4 TRANSPORT PRIORITY DESCRIPTIONS

The different types of transport priorities are listed below:

- **Emergency Transport** – System or process is down and significant impact to PRD or business process is occurring. This type of transport will be sent immediately upon approval by the appropriate manager or his/her designee.
- **End of Day** - System or process is being impacted, but the transport is not of an urgent nature and can wait to be corrected before the start of next business day.
- **Quality Transport in 2hrs**
- **Quality Transport in 30 minutes**
- **Bi-Weekly Transport**
- **Support pack to PRD**
- **Migration Freeze Attempt**
10.5. TRANSPORT REQUEST FORM

The New Transport form is attached below for reference. Additional details of the Transport process can be found here.

![New Transport form](image)

**Approver matrix:** Given below is a matrix of all the approvers and the Target client where they are authorized to approve the Transport Requests.

<table>
<thead>
<tr>
<th>Approvers</th>
<th>Team</th>
<th>First Approver</th>
<th>Final Approver</th>
<th>TRN</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mary Dolan*</td>
<td>HR</td>
<td>✓</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Fred Zupp</td>
<td>ABAP</td>
<td>✓</td>
<td>✓</td>
<td>-</td>
</tr>
<tr>
<td>Yong Park*</td>
<td>ABAP</td>
<td>✓</td>
<td>✓</td>
<td>-</td>
</tr>
<tr>
<td>Andy Duran**</td>
<td>ERP/ Portal/BW</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Norm Rille</td>
<td>BW</td>
<td>✓</td>
<td>-</td>
<td>-</td>
</tr>
</tbody>
</table>
**Bi-Weekly Transports**

Bi-Weekly transports are carried out once in a fortnight. It is a process where in the Final Production approval is automated, and the record owners can pick the week for production and also select the order in which the transports have to be done. A link to the Bi-Weekly transports documents can be found [here](#).
11. AUTHORIZATION STANDARDS

11.1 AUTHORIZATIONS CONCEPT

The architecture of LACCD’s SAP authorizations system is based upon the utilization of several individuals but related logical components: Profiles, Objects, Fields, and Authorizations. Users can only use those transactions and programs that they are explicitly allowed to access. The user ID refers exclusively to profiles. Each profile grants a set of specific system access authorizations to user.

Different groups must be created within the SAP team, like for ABAP developers, HR team, FI team, testing team.

Programs that can be executed by all LACCD users must be configured using the authorization group ‘ZABAPALL’. This configuration is done by the developer of the report program in the ‘Attributes’ screen of SE38.

Whenever a developer, tester or functional staff needs authorizations to a screen or t-code, they must follow the process of logging the request in Authorization_Request project of HP Quality Center. The request will then be approved by the Manager and the Authorizations team will do the needful.

Programs that should only be executed by a restricted set of users should have an authorization group that is defined to include only that set of users. (In many cases, it may be desirable for a large set of developers and testers to have that authorization in DEV, a smaller set to have that authorization in QAS, and an even more restricted set to have that authorization in production.)
Some programs have other authorization controls (for example, the program may provide particular functionality to certain users as determined by the ABAP "AUTHORITY-CHECK" command or an authorization checking function module), but other functionality is available to everyone. In that case the program should use the 'ZLACCDALL' authorization group in addition to its explicit use of the ABAP "AUTHORITY-CHECK" command or an authority checking function module.

In summary, all LACCD-written reports must have an explicit authorization group - either as the only authorization mechanism or in conjunction with some other authorization mechanism. The 'ZLACCDALL' authorization group should be used where the intent is to allow everyone in the LACCD to at least start the program (although another authorization mechanism may be used to further restrict who can perform particular functions or access particular data).

11.2 AUTHORIZATIONS: USAGE OF SAP OBJECTS

For any custom programming using authorization checks, developers must determine whether a standard SAP authorization object should be used, or if a custom object should be developed. Since the authorization object implementation typically comprises more business-related than technical issues, developers should consult with the Coordinators responsible for the application in making this decision.

11.3 RESTRICTING ACCESS TO HR DATA

Because of the sensitivity of HR data, all custom-written programs that access HR data must restrict who can access the HR data. Four methods to accomplish authorization checks for HR data are listed below:
Use function module **HR_READ_INFOTYPE** instead of direct SELECT statements when reading a specific infotype.

Use logical database to leverage SAP authorizations

SELECT statements should only be used when the SAP documented data interfaces which incorporate the SAP authorization checks (Logical Databases, function modules, and BAPI's) cannot provide the functionality required. If it is necessary to use SELECT's, then you must perform your own **AUTHORITY-CHECK** on the data selected.

Place a strict authorization group at the program (transaction) level. If a wide variety of data for a large group of individuals is needed in a single program, then this program must have a very strict authorization on who can run it.
12. TESTING STANDARDS

All project events and project success stem from testing and testing well. The purpose of this section is to describe the standards and procedures to follow during the testing phase of projects at LACCD.

HP Quality Center is the Test Management tool at LACCD. It is the repository for test design and test execution results. It can be accessed from the link given below:

http://mercury.laccd.edu/qcbin/start_a.htm
1. Enter your username (LACCD Email Username) and password (LACCD Email Password) and click Authenticate button.

2. Once the user details are authenticated, the user can select from the Domain Name and Project Name drop down boxes to login into the respective projects.

The project names and brief explanation for all projects that are in the Support Mode are given below:

**Domain Name: QC_MANAGEMENT**

<table>
<thead>
<tr>
<th>Project Name</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Authorizations_Requests</td>
<td>To request for authorizations, create new roles in SAP etc.</td>
</tr>
<tr>
<td>Transport_Requests</td>
<td>To request approvals to move Transports to other clients</td>
</tr>
<tr>
<td>Issue_Management</td>
<td>To log tickets by the end user and test management</td>
</tr>
<tr>
<td>Firecall_Request</td>
<td>To request for approval to open the production system for direct update or change.</td>
</tr>
</tbody>
</table>

Projects that are in implementation phase are created under ‘Projects’ domain. Access to these projects is limited to members of the team who are actively working on the project.

These standards and procedures may be changed via a change control mechanism that allows all those concerned to be notified of changes made to the steps. All projects (including enhancements) should be tested internally within IT before being reviewed by the requestor. Appropriate support teams in IT should be available to assist with this task.
12.1. TESTING PHASES

12.1.1 UNIT TESTING

This is the lowest level of testing at the SAP transaction level. All developers are responsible for planning their own test cases. Test case should include an appropriate sampling of data before and after project has run. They should also include measures to ensure the appropriate outcome. Each defect or issue logged in the Defects or Issue Module should have a corresponding test case associated with it. It should be linked to the issue. The test case should be successfully executed in the Test Plan before it is moved to QAS.

A program review should be conducted by the Team Manager.

- All return codes (sy-subrc) should be tested for success and failure after any I/O and calls to function modules (database selects, internal table reads, call transaction, etc.)
- The runtime analysis in the ABAP workbench should be conducted on the source code to get an overview of the duration and performance of the source code. The results of the analysis should be used as a basis to optimize the code.
- All projects should be tested/reviewed and signed-off by the requestor before they are moved to production.
- By the end of unit testing, Test Plan and Test Cases must be thoroughly documented in Quality Center for each Issue/Transport and Project.
**Unit Testing Types**

Unit testing should include boundary testing for positive and negative testing.

The configuration team owns the unit-testing effort and is responsible for planning and execution of unit testing. The main focus of unit testing is:

- Master data
- Negative-positive testing
- Transaction functionality
- Security roles and profiles

Unit testing should include testing security roles. Negative testing should be performed on security roles and profiles, custom fields, objects, and processes.

Each test in negative testing should have two elements:

- Intentionally specify conditions that will cause the software to generate an error.
- Ensure that the generated error is handled in a specified manner.

Negative testing examples:

- An example of a negative test for a process would be attempting to process an order with the wrong status.
- An example of a negative test condition would be "Attempting to post a material to an invalid profit center should produce an error message."
- Another negative testing example for security roles and segregation of duties would be "An inventory clerk attempts to approve a million-dollar purchase order when he is only permitted to approve purchase orders for a maximum of $500,000."
Negative testing should be designed to address the following situations:

- Check exception handling and error message.
- Prove that the system will deal with program exceptions and erroneous data.
- Limit or prevent an end user from trying to do something he should not.
- Demonstrate that the system does not do anything that it is not supposed to do.
- Users are permitted to perform only actions based on their authorizations, position roles, and permissions.

**Unit Test Procedure**

The procedure for unit testing is as follows:

- In the Test Plan module of HP QC, state the condition that will be tested by the test case (this should be used as the title of the test case)
- List the steps/actions to be performed in order to accomplish the test as Design steps within the test case
- For each action performed, identify the expected result
- Create test data necessary to create the condition being tested and for each piece of data, indicate the expected results
- Run the tests in the Test Lab module of Quality Center and log the results
- In case there is a defect, log the defect in the Defects module of Quality Center
- Re-run the necessary tests after the defect is fixed
- Change the status of the defect in QC as fixed
- Package the test documentation and pass it to the First Approver.

The First Approver will verify that the documentation in Quality Center is complete, review the test case and the code.
Unit Test Case Template

IEEE Standard 610 (1990) defines test case as follows:

“(1) A set of test inputs, execution conditions, and expected results developed for a particular objective, such as to exercise a particular program path or to verify compliance with a specific requirement.”

For SAP testing, test cases should contain, at a minimum, the following information and fields:

- SAP roles needed to execute a process(es) (i.e., warehouse clerk).
- Data value(s) or data variants needed to execute the test (i.e., enter data value "1000" for company code, test process with multiple "wage types" such as straight time, holiday time, overtime, vacation time, etc.).
- Requirement met or fulfilled from executing the test case.
- Any preconditions that are needed for executing a test case (i.e., a requisition is needed before generating a purchase order).
- Description of the process to be tested.
- Approval fields (i.e., for signoffs).
- Test steps to be performed.
- Expected test results\run.
- Actual test results (i.e., pass or failure).
A good test case should contain the following elements as shown below:

<table>
<thead>
<tr>
<th>Test Case Id</th>
<th>Descriptive of the process</th>
<th>SAP Role</th>
<th>Test Created By</th>
<th>Prerequisites</th>
<th>Test Procedure</th>
<th>Test Data</th>
<th>Expected Result</th>
<th>Actual Result</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>Serial no assigned to test case – automatically assigned</td>
<td>Brief idea about case</td>
<td>SAP Role needed to execute the test case</td>
<td>Name of test creator</td>
<td>Conditions that should be fulfilled before the test is performed</td>
<td>Steps to be performed in test</td>
<td>Inputs, variables and data</td>
<td>What the program should do</td>
<td>What is actually done. Whether results are as expected or not.</td>
<td>Notes on the procedure</td>
</tr>
</tbody>
</table>
Example of a well written test case is shown below:

Each step should be clearly defined and easy to be followed. What action should be taken is explained in the ‘Description’ column and the results expected are mentioned in the ‘Expected’ column. The data needed is mentioned in the Description steps.

<table>
<thead>
<tr>
<th>Step Name</th>
<th>Description</th>
<th>Expected</th>
</tr>
</thead>
<tbody>
<tr>
<td>Step 1</td>
<td>Execute transaction IL01</td>
<td>Initial Screen is Displayed Create functional location initial screen</td>
</tr>
<tr>
<td>Step 3</td>
<td>Create functional location master data screen Default data from copy from and superior functional location populated the following fields: 1. MaintPlant- &quot;M&quot; 2.Work Center- &quot;1000&quot; 3. Planing Plant - &quot;M&quot; 4. Planner Groups- &quot;100&quot; 5. Main Work Center- &quot;1000/M&quot; The user has the option to enter the following values: Object Type - &quot;Zone&quot; Sort Field - &quot;M-Z1&quot; Enter to validate the option values input. Click on the tab titled &quot;List of Equipment&quot;</td>
<td>The system accepts the entered values and opens the list of Equipment tab to allow the user to enter data</td>
</tr>
</tbody>
</table>
| Step 4 | Enter the following data into the "list of Equipment" tab as shown below:
Asset: "blank" if the functional location has an asset and asset sub-number, the user can enter the asset and sub asset codes in this field
In the equipment assignment window, the user can click on the right hand side sub equipment icon to open the "install Equipment"

Enter the equipment number if known, or click on the dropdown icon to open up the equipment search to locate the equipment that needs to be installed at this functional location. The user would be also required to enter the installation position, and date and time of the installation

The system accepts the entered values, and the data is ready for save to conclude the creation of the functional location. |
| --- | --- |
| Step 5 | Click on the save icon to save the entered data to create the functional location.

The system generates a message advising the user that the functional location has been created successfully. |
Test Case: Naming Convention

LACCD’s Quality Assurance department is responsible for defining and implementing the Test case naming standards.

Goal of naming standards:

- Meet Quality Assurance standards
- Consistency in maintaining test cases
- Ease to search required test cases
- Required data and test cases can be easily searchable by different personnel of any other department.
- Convenient to enhance or reuse required test cases
- Establish company standards in maintaining test cases
Test cases should be named as per standards given below:

<table>
<thead>
<tr>
<th>Test Case</th>
<th>Naming Standard</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>Functional Test case</td>
<td>Module Area-Sub Module-Functionality tested where, Module Area and Sub Module is represented by two letter abbreviation. Functionality is a phase that describes a SAP process.</td>
<td>HR-PA-New Hire</td>
</tr>
<tr>
<td></td>
<td></td>
<td>FI-AM-Create WBS element</td>
</tr>
<tr>
<td>Issue related Test case</td>
<td>Issue # - Module Area-Sub Module-Functionality tested Where, Issue# is the QC issue#, Module Area and Sub Module is represented by 2 letter abbreviation. Functionality is a phase that describes a SAP process.</td>
<td>#2548-HR-TM-New work schedule</td>
</tr>
<tr>
<td></td>
<td></td>
<td>#3145-FI-MM-Invoice Verification</td>
</tr>
<tr>
<td>Integration-Test case</td>
<td>To be decided</td>
<td></td>
</tr>
</tbody>
</table>

Please click [here](#) to learn more about creating test cases, running tests and logging defects in HP QC in the HP QC job aid.

**Business Warehouse Testing**

Some points to keep in mind when testing BW reports are given below:

- Reconciliations - Are financial calculations rolling up correctly?
- Extracts - Is there a match between the number of extracted records and the number of received records.
- Performance - How fast can a query be performed, and does it
- Security - Who is permitted to slice and dice the data in the BEx Analyzer? What are the established roles for generating the queries?
- Data Transfer rules - Is data transformed correctly for all fields from the source system to the target system.
12. Testing Standards

12.1.2 SCENARIO TESTING

Scenario testing is the testing of chains of SAP transactions that make up a process within a single area or module. It will include testing of the process with data from external systems and applicable SAP roles.

It is owned by the configuration teams but includes participation of SME’s and members of the test and development team.

12.1.3 INTEGRATION TESTING

It is the testing of chains of SAP transactions that make up an end-to-end process that cuts across multiple modules. For instance hire-to-retire and purchase-to-pay with external data and converted data. The goal of integration testing is to ensure that all interacting subsystems in a system interface correctly with one another to produce the desired results. These tests ensure that the introduction of one or more subsystems into the existing system does not have an adverse effect on existing functionality.

Integration testing will require participation from members of the configuration and development team for defects resolution. Additionally, SME’s and end users should participate in the integration test as reviewers and for approval of test results.

Integration Testing consists of:

- Creating the integration test plan
- Creating test data
- Conducting tests according to the integration test plan
- Reporting and reviewing the results of the test run
12.1.4. PERFORMANCE TESTING

Performance testing will encompass load, volume, and stress testing to determine special bottlenecks and degradation points. A dedicated test team is the owner of the performance test. Performance testing is primarily done with automated testing tools, but will still include manual execution of interfaces, batch jobs, and external processes that send data into SAP.

The basis, database and infrastructure team should help to monitor the performance test, whereas the configuration team should help to identify test data and document test cases that are suitable for performance test.

12.1.5. USER ACCEPTANCE TESTING

User Acceptance Testing allows the system end users to independently execute the test cases from the perspective of how the end users plan to perform tasks in the production environment. The owners are end user and the configuration and test team members resolve defects identified during the user acceptance test. The test team and change management team members help train end users and prepare them for the user acceptance test.

12.1.6. REGRESSION TESTING

Regression testing ensures that previously working system functionality is not adversely affected by the introduction of the new system changes. System changes targeted for the production environment need to be analyzed for impact and cascading effects on other processes. Regression testing is primarily an automated testing effort. Determining the impact of system change is primarily the responsibility of the integration team and upper management.
13. ISSUE MANAGEMENT STANDARDS

This chapter outlines the instructions to follow for logging issues and monitoring the progress of the issue as it moves from ‘New’ to ‘Closed’ status. Issues can be added any time by authorized users.

LACCD uses the term ‘Issue’ when the problem occurs in a production environment, and the term ‘Defect’ is used to indicate a deviation from the required behavior during the implementation of a new module. Issues are logged in ‘Issue_Management’ project within the ‘QC_MANAGEMENT’ domain. Defects are logged in the respective projects under implementation.
LOGGING ON HP QUALITY CENTER

1. Once the user details are authenticated as mentioned in the previous chapter, select domain name as ‘QC_MANAGEMENT’ and project name as ‘Issue_Management’.

2. Click Login. The Defects Grid view will be opened as shown below:

<table>
<thead>
<tr>
<th>Issue ID</th>
<th>Issue Title</th>
<th>Module/Area</th>
<th>Baseline D</th>
<th>Priority</th>
<th>Date Resolved</th>
<th>Issue Status</th>
<th>Primary Owner</th>
<th>Reporting Comments</th>
<th>Problem Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>2473</td>
<td>Schedule to Reduce Basic Life Cover Human Resources</td>
<td>2207/2009</td>
<td>3-High</td>
<td>32/02/09</td>
<td>Open</td>
<td>Resolved Bug</td>
<td>John Doe</td>
<td>Maintainer: John Doe</td>
<td>Design</td>
</tr>
<tr>
<td>2503</td>
<td>EASY VDI for all users</td>
<td>2207/2009</td>
<td>3-High</td>
<td>11/10/09</td>
<td>High</td>
<td>Resolved Bug</td>
<td>Jane Smith</td>
<td>Maintainer: Jane Smith</td>
<td>Bug</td>
</tr>
<tr>
<td>2508</td>
<td>Ensure that when 27A is inactive a bug</td>
<td>2207/2009</td>
<td>2-Very High</td>
<td>22/12/09</td>
<td>Open</td>
<td>Resolved Bug</td>
<td>Sarah Lee</td>
<td>Maintainer: Sarah Lee</td>
<td>Bug</td>
</tr>
<tr>
<td>2611</td>
<td>Production Support: FMIHPRF</td>
<td>11/05/2012</td>
<td>1-Urgent</td>
<td>12/06/2013</td>
<td>Assigned</td>
<td>def if the payrol Integrator Maintenance and</td>
<td>John Doe</td>
<td>Maintain: John Doe</td>
<td>Bug</td>
</tr>
<tr>
<td>2855</td>
<td>Archiving: Not used for Material Management</td>
<td>11/06/2013</td>
<td>1-Urgent</td>
<td>22/12/2013</td>
<td>Assigned</td>
<td>def if the payrol Integrator Maintenance and</td>
<td>Jim Brown</td>
<td>Maintain: Jim Brown</td>
<td>Bug</td>
</tr>
<tr>
<td>3197</td>
<td>Benchmark Update: Human Resources</td>
<td>11/06/2013</td>
<td>1-Urgent</td>
<td>22/12/2013</td>
<td>Assigned</td>
<td>def if the payrol Integrator Maintenance and</td>
<td>John Doe</td>
<td>Maintain: John Doe</td>
<td>Bug</td>
</tr>
<tr>
<td>3292</td>
<td>Create Online TV Document</td>
<td>11/06/2013</td>
<td>2-Very High</td>
<td>22/12/2013</td>
<td>Assigned</td>
<td>def if the payrol Integrator Maintenance and</td>
<td>Jim Brown</td>
<td>Maintain: Jim Brown</td>
<td>Bug</td>
</tr>
<tr>
<td>3381</td>
<td>Resolve: Service: SAP Performance Manager</td>
<td>11/06/2013</td>
<td>3-High</td>
<td>12/06/2013</td>
<td>Assigned</td>
<td>def if the payrol Integrator Maintenance and</td>
<td>John Doe</td>
<td>Maintain: John Doe</td>
<td>Bug</td>
</tr>
<tr>
<td>3333</td>
<td>Module: 2: Dispute Subtype, not Human Resources</td>
<td>11/06/2013</td>
<td>4-Medium</td>
<td>22/12/2013</td>
<td>Assigned</td>
<td>def if the payrol Integrator Maintenance and</td>
<td>Jim Brown</td>
<td>Maintain: Jim Brown</td>
<td>Bug</td>
</tr>
<tr>
<td>3409</td>
<td>User Support: 2: SAP Support Human Resources</td>
<td>11/06/2013</td>
<td>4-Medium</td>
<td>22/12/2013</td>
<td>Assigned</td>
<td>def if the payrol Integrator Maintenance and</td>
<td>John Doe</td>
<td>Maintain: John Doe</td>
<td>Bug</td>
</tr>
<tr>
<td>3453</td>
<td>Payroll: User support</td>
<td>11/06/2013</td>
<td>2-Medium</td>
<td>22/12/2013</td>
<td>Assigned</td>
<td>def if the payrol Integrator Maintenance and</td>
<td>Jim Brown</td>
<td>Maintain: Jim Brown</td>
<td>Bug</td>
</tr>
</tbody>
</table>

13.2 NEW ISSUE CREATION

A new issue is created whenever the end user calls the help desk to report problems in production. An issue is also created whenever support packs are applied, during a GUI upgrade, configuration changes, or a new module has been added. At the time of implementation of a new project, a defect is created whenever, the actual test results are different from the expected test results documented within a test case.

Click on ‘New Issue’ button to open the New Issue dialog box. Alternately, click on Issues -> New Issue to open the New Issue Dialog box. The Create ‘New Issue’ dialog box is shown below:
13. Issue Management Standards

‘Issue Details’ Tab in New Issue Dialog Box

A brief description of the fields in the Issue Details Tab of the New Issue dialog box is given below:

1. **Issue Title**: Enter brief details of the Issue here.
2. **Module / Area**: Select the appropriate Module where the issue is found. The options are as given below:
   
   - Basis
   - Business Warehouse
   - Finance
   - Government Risk and Compliance
   - Human Resources
   - Institutional Effectiveness System
3. **Sub Module**: Depending on the Module selected, select the relevant Sub Module from the Sub Module drop down. The various options are given below:

<table>
<thead>
<tr>
<th>Module/Area</th>
<th>Sub Module</th>
</tr>
</thead>
<tbody>
<tr>
<td>Basis</td>
<td>Basis</td>
</tr>
<tr>
<td>Business Warehouse</td>
<td>BI Portal Integration</td>
</tr>
<tr>
<td></td>
<td>Data Modeling</td>
</tr>
<tr>
<td></td>
<td>Extracts</td>
</tr>
<tr>
<td></td>
<td>Reports</td>
</tr>
<tr>
<td>Finance</td>
<td>Accounts Payable</td>
</tr>
<tr>
<td></td>
<td>Accounts Receivable</td>
</tr>
<tr>
<td></td>
<td>Asset Management</td>
</tr>
<tr>
<td></td>
<td>Budget</td>
</tr>
<tr>
<td></td>
<td>Contracts</td>
</tr>
<tr>
<td></td>
<td>eBTA</td>
</tr>
<tr>
<td></td>
<td>Funds Management</td>
</tr>
<tr>
<td></td>
<td>General Accounting</td>
</tr>
<tr>
<td></td>
<td>General Ledger</td>
</tr>
<tr>
<td></td>
<td>Grants Management</td>
</tr>
<tr>
<td></td>
<td>Position Budget Control</td>
</tr>
<tr>
<td></td>
<td>Project Systems</td>
</tr>
<tr>
<td></td>
<td>Public Budget Formulation</td>
</tr>
<tr>
<td>Government Risk and</td>
<td>Access Control</td>
</tr>
<tr>
<td>Compliance</td>
<td>Budget Control</td>
</tr>
<tr>
<td></td>
<td>Process Control</td>
</tr>
</tbody>
</table>
| Human Resources          | Benefits  
|                         | EASY  
|                         | Employee Self Services  
|                         | eRPA  
|                         | Manager Self Services  
|                         | Organizational Management  
|                         | Personnel Administration  
|                         | Personnel Change Request  
|                         | Retirement  
|                         | Time Management  
| Institutional Effectiveness System | Institutional Effectiveness System  
| Materials Management | Goods Receipt  
|                         | Invoice Verification  
|                         | Purchasing  
| Others | HR-FI Integration  
|         | Change Management  
|         | Business Process  
| Payroll | BSI  
|         | Garnishments  
|         | Payroll  
|         | Payroll FI Posting  
|         | Retirement  
|         | Tax Reporter  
|         | Third Party Remittance  
| Plant Maintenance | Billing Documents  
|         | Capacity Planning  
|         | Completion Confirmation  
|         | Maintenance Plan  
|         | Measuring Documents  
|         | Notifications  
|         | PM Master Data  
|         | Work Orders  
| Portal | Benefits  
|         | Employee Self Services  
|         | Manager Self Services  
|         | Reports  
|         | Time  
| Protocol | Protocol  
|
4. **Baseline Date**: Select the approximate date when the issue must be fixed from the Baseline Date drop down.

5. **Problem Type**: Depending on the type of anomaly, the issue can be classified into various categories. Please select the appropriate category of problem that fits the issue on hand.

<table>
<thead>
<tr>
<th>Classified as</th>
<th>if...</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Basis</strong></td>
<td>Issue related to basis.</td>
</tr>
<tr>
<td><strong>Break</strong></td>
<td>There is an unexpected defect in the system's functionality.</td>
</tr>
<tr>
<td><strong>Bug</strong></td>
<td>Any deviation from the business requirements.</td>
</tr>
<tr>
<td><strong>CE Related</strong></td>
<td>n/a</td>
</tr>
<tr>
<td><strong>Design</strong></td>
<td>Issue related to the design of the project</td>
</tr>
<tr>
<td><strong>Enhancement</strong></td>
<td>Issue is an enhancement to the existing functionality.</td>
</tr>
<tr>
<td><strong>Functional</strong></td>
<td>Issue related to functional working of the program (redundant – since we can classify it as a bug or a break).</td>
</tr>
</tbody>
</table>
6. **Company code** – Select the appropriate company code from the Company Code drop down box.

7. **Issue Status** – The status of the issue will be **New/Open** when an issue is added into the system.

8. **Priority** – Depending on the importance and urgency of resolving the issue, choose the appropriate option from the Priority drop down. The options in the drop down box are as below:
   - 1-Urgent – Issue is a show stopper and requires immediate resolution issue has occurred in a production system or an imminent go-live/upgrade is jeopardized.
   - 2-Very high - It has serious consequences for business operations and requires a resolution within one to three days.
- 3-High – Business operations are seriously threatened and urgent task cannot be executed
- 4-Medium - Business operations are affected; however, there is a workaround to the issue.
- 5-Low – Issue does not hinder daily operations and has little influence on daily operations.
- 6-Not Prioritized

**Note:** Any issue that is under Development/Testing should have a priority level defined. It cannot remain in the ‘Not Prioritized’ status.

The proposed turnaround time to resolve a defect is given below:

<table>
<thead>
<tr>
<th>Priority</th>
<th>Resolved within</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 - Urgent</td>
<td>As soon as possible, within 1 business day.</td>
</tr>
<tr>
<td>2 - Very High</td>
<td>As soon as possible or within 1-3 business days</td>
</tr>
<tr>
<td>3 - High</td>
<td>Within 4 business days</td>
</tr>
<tr>
<td>4 - Medium</td>
<td>Within 1-2 work weeks</td>
</tr>
<tr>
<td>5 - Low</td>
<td>If within scope resolve at any time within existing release or future release.</td>
</tr>
</tbody>
</table>

9. **Primary Developer** – Select the name of the team member who is the main person responsible for getting the issue fixed.

10. **Sub System Impacted** – Select the name of the sub system which has been impacted by the issue.

11. **Difficulty** – Select the degree of difficulty that can be defined in fixing the issue.
12. **Functional User Importance** - This is up to the discretion of the SME or the Manager to decide if the issue belongs to the Top 10 or Top 12 category. If the number of issues in the particular category exceeds the limit, no new issues can be added in this category. The user will be given a message as below:

![Message from Quality Center](image)

13. **Business Sponsor** – Select the name of the business sponsor for the issue being created.

14. **Date Needed** – The actual date by which the issue has to be fixed. If it is blank, it will be auto-populated with the baseline date.

15. **Description Field** – Enter the steps to reproduce the issues, or give a brief description that can help the developer to resolve the issue.

‘Tasking Tab’ in New Issue Dialog Box

When a new issue is created enter the names of the respective team members who are assigned to work on the issue as shown in the Tasking Tab below:
Notes/Solution Tab

Any additional details and note can be entered here.

Click ‘OK’. The issue will be added to the system and a new ‘Issue Id’ will be automatically assigned to it by the system.
13.3 ISSUE DETAILS DIALOG BOX

To view the details or to enter additional details in an already logged issue double click on the issue in the Issue Details grid to open the Issue Details dialog box.

Alternately, if the Issue ID is known, searching for a particular issue can be done by clicking on Issues -> Go to Issue in the menu bar. The Issue Details dialog box will be opened as shown below.

‘Details Tab’ in Issue Details Dialog Box
The mandatory fields are highlighted in red, all other fields are optional. The ‘Primary Developer’ field is made a mandatory field in the Issue Details box.

Also, any communication or comments can be added by clicking on the ‘Add Comment’ button in the Issue Details dialog box. Previous comments or communication cannot be deleted. Clicking on ‘OK’ button will save all the changes that will be made to the issue.

**Attachments:** Any additional/supporting documents related to the issue can be attached by clicking on the ‘Attachments’ icon in the side menu bar.

**Linked Entities:** Any of the issues that have been created should be tested before transporting to production. This is done by creating a test case and linking the issue to the test case. Please refer to [Unit Testing Framework document](#) to learn how to link the defect to the test case.

**History:** Clicking on this icon, will open the audit record of all the modification made on the issue.

**‘Tasking/time’ tab in Issue Details Dialog Box**

In the tasking/time tab select the appropriate name of the person who is assigned as SAP team member, consultant, functional staff, additional staff etc., If ABAP or Basis work is needed enter the name of the team member who is responsible to work on the issue.
13. Issue Management

**SAP Team Time (hrs)** – Whenever the member who has been assigned to the issue as indicated in the SAP Team dropdown box works on the issue, he/she should enter the time spent on that issue each day they work on the issue.

**Consultant Time (hrs)** – Whenever the consultant works on the issue, he/she needs to update the Consultant Time for that day.

**Functional Time (hrs)** – If the issue is fixed by the Functional Staff, enter the time spent to fix the issue on that day.

**Addl. Staff: Time (hrs)** – If the issue is fixed by the Additional Staff, enter the time spent to fix the issue on that day.
ABAP Assig 1: Time (hrs) - When there is ABAP work involved, the assigned person should enter the time spent on the issue on that day.

ABAP Assig 2: Time (hrs) - same as above.

Additional Personnel – If additional work is needed by any other team member, then select the name of the team member.

Work Area – If the Additional Personnel field is not blank, select the work area that the additional person will be working on.
‘Time Logged’ tab in Issue Details Dialog Box

The time logged tab will contain the cumulative time that is logged by the various team members for that issue. The date when the issue was closed is shown in the ‘Date Resolved’ field. The number of days to resolve the issue is indicated in the ‘Elapsed time (in days)’ box. All the fields in this tab are read only fields.
13.4 EMAIL NOTIFICATIONS

Email notification is sent to users whenever the following fields are modified:

- ABAP Assigned #1
- ABAP Assigned #2
- Additional Staff
- Basis Assigned: #1
- Basis Assigned: #2
- Communication History
- Consultant
- Delivery Status
- Department Name
- Dept. Manager
- Functional Staff
- Issue Status
- Primary Developer
- Priority
- SAP Team

The email subject for the email notifications will be as shown below:

*Issue # XXXX related to <Module Area> has been created/updated - <Issue Status>*

Where XXXX stands for Issue Number.
13.5 VIEWING ISSUES FROM THE EMAIL

Upon receiving the email click on the hyperlink and logon to Quality Center.

After logging in, the exact issue will be highlighted in the grid in Quality Center. Open the Issue by double clicking on the issue record or click on ‘Issue Details’ button.
13.6 SEARCH FOR AN ISSUE BY ISSUE TITLE

You can use text search to search for a particular issue. Click the **Text Search** button or choose **Edit > Text Search**. The text search pane opens in the lower part of the window.

1. In the **Search for** box, type the words you want to find. If you are working in the Test Plan module: In the **In** box, select **Tests** or **Design Steps**.
2. To search all records in the module, clear the **Restrict to current filter** check box.
3. Click **Search**. Quality Center performs the text search on the predefined fields and displays the search results in order of relevance.
4. To change the column appearance and order, click the **Select Columns** button. The Select Columns dialog box opens. For more information, see Arranging Columns.
5. To view the list of predefined search fields set in Project Customization, click the **Searchable Fields Information** button. The list of predefined search fields is displayed. Click **OK** to close the searchable fields list.
6. To display record details, select a record and click the record **ID** or **Name** link. Alternatively, select the record and click the **Go to Entity** button.

<table>
<thead>
<tr>
<th>Defect ID</th>
<th>Status</th>
<th>Priority</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
13.7 ISSUE TRANSITION

Issue transition standards have been created to streamline the process of how the issues will move from one state to another.

The possible states of the issues created will transition from one state to another as per the Issue Transition Matrix given below:

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Open</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Assigned</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Being Analyzed</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Dev/Coding</td>
<td>x</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Testing</td>
<td></td>
<td>x</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>User Acceptance Testing</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Avgt User Sign-Off</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Monitoring</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Avgt Bus Spec/Avgt Doc</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Delayed for Future Improvement</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Inactive</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

X indicates status will be automatically updated if issue is lying in current state for more than 90 days.

X indicates status will be automatically updated if issue is lying in current state for more than 365 days.

X indicates change to the new status will require management approval.

It is important to flag the issue status to match the state of workflow. Issues in the ‘From State’ column can only move to ‘To State’ along the rows where an ‘X’ has been marked.

For example, Issue status can be changed from

‘Open’ to ‘Assigned’, or
‘Testing’ to ‘User Acceptance Testing’, or
‘Being Analyzed’ to ‘Cancelled’ or ‘Awaiting Business Documentation’.
However, if a user tries to change the issue status from Open to Dev/Coding then the change will not be allowed and the following message will be displayed.

Issue Status **cannot** be changed from for example:

- ‘Open’ to ‘Development/Coding’, or
- ‘Development/Coding to ‘Monitoring’, or
- ‘Testing’ to ‘Assigned’, or
- ‘Inactive’ to ‘Development/Coding’

If a user tries to make an invalid issue transition the following message will be given and action will be denied

![Message from Quality Center](image)

**Aging Issues**

Aging issues will be managed as per the below guidelines:

<table>
<thead>
<tr>
<th>Current Issue Status</th>
<th>Rule</th>
<th>Change To</th>
</tr>
</thead>
<tbody>
<tr>
<td>Awaiting Business Specifications</td>
<td>&gt; 90 days</td>
<td>Delayed for Future Improvement</td>
</tr>
<tr>
<td>Awaiting Documentation</td>
<td>&gt; 90 days</td>
<td>Delayed for Future Improvement</td>
</tr>
<tr>
<td>Assigned</td>
<td>&gt; 365 days</td>
<td>Delayed for Future Improvement</td>
</tr>
<tr>
<td>Open</td>
<td>&gt; 365 days</td>
<td>Delayed for Future Improvement</td>
</tr>
<tr>
<td>Being Analyzed</td>
<td>&gt; 365 days</td>
<td>Inactive</td>
</tr>
<tr>
<td>-------------------------------</td>
<td>------------</td>
<td>------------</td>
</tr>
<tr>
<td>Delayed for Future Improvement</td>
<td>&gt; 365 days</td>
<td>Inactive</td>
</tr>
<tr>
<td>Development/Coding</td>
<td>&gt; 365 days</td>
<td>Inactive</td>
</tr>
<tr>
<td>Testing</td>
<td>&gt;365 days</td>
<td>Inactive</td>
</tr>
<tr>
<td>Monitoring</td>
<td>&gt; 90 days</td>
<td>Closed</td>
</tr>
</tbody>
</table>

For example, any issues that are currently in Awaiting Business Specification state for more than 90 days will be automatically changed to ‘Delayed for Future Improvement’ status.

**Management Approval**

Any issues in ‘Inactive’ or ‘Delayed for Future Improvement’ status will require management approval to be re-opened.

If a user tries to change an issue that is in Inactive or Delayed status he/she will receive the following message:

Issues in Inactive or Delayed Status can only move to ‘Inactive – Pending Management Approval’, or ‘Delayed – Pending Management Approval’ respectively.
Given below is the explanation of the issue status:

**New/Open** – This is the status when an issue is added into the system.

**Assigned** – The developer or the Team Lead will change the status to Assigned and notification is sent to the functional/ABAP team member.

**Being Analyzed** – The SAP team member to whom the issue is assigned will now change the status to 'Being Analyzed' – while doing research on the issue, analyzing data etc.,

**Dev/Coding** - The SAP team member working on the issue will change it to Dev/Coding.

**Testing** - After coding/configuration is completed the status should be changed to Testing.

**User Acceptance Testing** - After the issue is tested and passed, the Team Member should change the status to User Acceptance Testing – it will indicate to the business user that the issue is ready for Testing.

**Awaiting User Sign-Off** - When the issue is related to production, then status is changed ‘Awaiting User Sign-Off’.

**Fixed** – When the issue is related to a project under development then change the status to Fixed

**Monitoring** – After sign-off, the issue will be in Monitoring status for next 90 days.

**Closed** – After 90 days in Monitoring Status the Production issue will be automatically changed to Closed status.
Awtg Documentation/ Awtg Business Specs - The issue can be changed to this status anytime in Being Analyzed or in Dev/Coding status whenever there is a need for additional clarifications from Business.

Cancelled – The issue can be updated to this status if the Issue was not an issue, or if the issue has been combined with another issue.

Combined – The issue can be updated to this status whenever another issue has been added to it, which are similar or related.

Delayed for Future Improvement – When the issue remains in Awaiting Business Specification/Awaiting Documentation, Assigned, Dev/Coding or New/Open status for more than 365 days, it moves to this status.

Inactive – The issue is moved to inactive when it is in Testing, Dev/Coding, Being Analyzed or Delayed for Future Improvements for 365 days. If the issue has to be opened again, it will require management approval.
14. REQUIREMENTS MANAGEMENT

Requirements are critical because they tell what the ‘system will do’, and the test team should develop test cases to ensure all requirements are verified and validated. Users can specify requirements within the Quality Center project’s Requirements module, which is opened by clicking the Requirements button on the sidebar.

Requirements module of HP QC should be used to determine the overall requirement for the application under development. Some of the key capabilities of Requirements Module in HP QC are as below:

- It can track multiple requirements types and analyze impact when requirements change.
- Requirements Traceability Matrix provides full traceability from requirements to tests and defects
- Leverage existing assets in MS Word/MS Excel: Existing defects in MSWord or MS Excel can be directly uploaded with the help of addins.

Requirements can be organized into folders as per logical functions and grouped under the following categories:

<table>
<thead>
<tr>
<th>Requirement Type</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Functional Requirements</td>
<td>Relate to tasks that the system's intended end users must perform in order to complete their job functions.</td>
</tr>
<tr>
<td>Security Requirements</td>
<td>Relate to segregation of duties, roles, profiles, and security authorizations that need to be capture and verified as testable requirements.</td>
</tr>
</tbody>
</table>
Development Objects | Relate to reports, interfaces, conversions, enhancements, forms (RICEF) and workflow specifications.

Usability Requirements | Requirements that assess the product’s UI technical environment, common user activities, navigation and interaction design and information design. The primary focus of usability is the end user and his or her tasks.

Govt. Regulated Requirements | Requirements that need to be verified to comply with the government regulations at the Federal, State, County or District levels.

The above requirements can be organized under folders and can be used to view the test coverage, and perform risk assessment.

### 14.1 ADD A REQUIREMENT

To add a new requirement, click the ‘New Requirement’ button. Alternatively, choose Requirements > New Requirement. The ‘Create New Requirement’ dialog box will appear as shown below. Enter an appropriate name:
Click OK and the 'New Requirement' dialog box will be displayed. Enter the Description in the Memo field below; assign the requirement a Priority level as discussed in the Test Plan, and other details.

Save the new requirement. A new requirement ID will be generated. User can view the requirement by clicking on the requirement in the display grid or by searching with the Requirement ID.
After defining your requirements, you can add traceability between the requirements. When analyzing the impact of a change proposed in a specific requirement, traceability shows the other requirements that the change might affect.

After adding the requirements they should be reviewed by the business sponsor and change the requirement from a ‘Not Reviewed’ status to ‘Reviewed’ status.

The user can also import requirements to the Quality Center project from Microsoft Word, Excel, or other third-party requirement management tools. To import requirements the appropriate HP Quality Center add-in must first be installed.
14.2 REQUIREMENTS TRACEABILITY MATRIX

No requirement is an island in itself; they will always be a parent-child relationship. So parent child relationship has to be defined between the requirements. **Requirements traceability** defines a relationship between the requirements. When analyzing the impact of a change proposed in a specific requirement, the traceability links indicate the other requirements that the change might affect. Any change in the parent requirement will have subsequent impact on the child requirement.

- The relationship between two requirements can be defined as follows:
  - Trace To - How will the change to a specific requirement affect other requirements?
  - Trace From – What requirements when changed will affect a specific requirement?
**Trace To** – links indicate requirements that are affected by a selected requirement.

Ex: Change to Faculty 3% requirement will impact Option B – Gross Roll Back requirement

---

**Trace From** – links indicate requirements that affect a selected requirement

Ex: Any change in Update Payroll Schema will affect Update Reports.
From Requirement Details view, select a requirement from the requirements tree. Click on the Requirements Traceability tab -> Impact Analysis Tab. Impact Analysis tab helps to understand the many associations and dependencies existing between the requirements by displaying them in a **Hierarchical Tree Structure**.

Traceability matrix is created by tracing requirements with the test cases or scenarios that verify them. The coverage analysis will give the user a snapshot of the test status by requirements coverage, as shown in the picture below:
The Test coverage of the requirements is explained as below:

- **Not Covered** – No Test is linked to requirement
- **Failed** – Test/s covered by the requirement has an execution status of failed.
- **Not Completed** – Test/s covered by the requirement has not been completed.
- **Passed** – Test/s covered by the requirement has passed
- **No Run** – Test/s covered by the requirement have an execution status of “No Run”
14. 3 RISK BASED QUALITY MANAGEMENT (RBQM)

Risk-based quality management can assist in determining testing strategy for the project requirements.

The elements that should be taken into consideration are:

**Risk Assessment** - Requirements that are children of analysis requirements and at a lower level in the requirements tree hierarchy.

<table>
<thead>
<tr>
<th>Risk Assessment</th>
<th>Functional Complexity</th>
<th>Analysis Results</th>
</tr>
</thead>
<tbody>
<tr>
<td>Calculated Risk:</td>
<td></td>
<td>Use custom Risk:</td>
</tr>
<tr>
<td>Business Criticality</td>
<td></td>
<td>Analysis Results:</td>
</tr>
<tr>
<td>Calculated Failure Probability:</td>
<td>Use custom Failure Probability:</td>
<td></td>
</tr>
<tr>
<td>Estimated development time (optional):</td>
<td>Hours</td>
<td></td>
</tr>
</tbody>
</table>

**Risk Analysis** - Risk Analysis is done on requirement at higher levels in the requirements tree hierarchy, such as the Folder type.

The Risk Category is composed of its Business Criticality and Failure Probability. The Functional Complexity Category indicates the complexity of the requirement’s
implementation. Business criticality measures how critical a requirement is for the application.

For example, a requirement affecting a minor feature that is likely to be used rarely might be assigned a Nice to Have Business Criticality, whereas a requirement that is essential to the application’s functionality would probably be assigned a Critical Business Criticality. Based on certain criteria QC will assign whether a requirement is

- A - Critical,
- B - Important, and
- C - Nice to Have

A requirement with a high Functional Complexity generally requires more testing time as it is more likely that the requirement’s implementation contains defects.
15. DOCUMENTATION STANDARDS

Given below are the LACCD document templates:

<table>
<thead>
<tr>
<th>Document</th>
<th>Link</th>
</tr>
</thead>
<tbody>
<tr>
<td>Project Plan</td>
<td>LACCD SAP Project Plan</td>
</tr>
<tr>
<td>Program Document</td>
<td>ABAP Program documentation</td>
</tr>
<tr>
<td>Production Control Document</td>
<td>Production Control Documentation</td>
</tr>
<tr>
<td>Change Management Document</td>
<td>Change Management Documentation</td>
</tr>
</tbody>
</table>

Note that a "general purpose" include file, or any include file which is not documented in the documentation for an associated main program, should be documented using the "program" template rather than the degenerate "include file" template. There must be documentation for the code within an include file either in the include file documentation itself, or in the main program's documentation.

It is mandatory to complete a documentation template prior to transporting your program to the production system.

In addition to documentation via the templates, brief comments within the code can be very helpful for ongoing program maintenance.
# 15.1 Naming Convention for Documents

<table>
<thead>
<tr>
<th>Document Type</th>
<th>Naming Convention</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>Project Plan Document (PP)</td>
<td>PP_modulearea_projectname</td>
<td>PP_PY_Adjunct Salary, which is Project Plan Document for Adjunct Salary in Payroll module</td>
</tr>
<tr>
<td>Program Document (PR)</td>
<td>PR_ModuleArea_ProgramName</td>
<td>PR_HR_NewHire, which is Program Document for NewHire program of HP Module.</td>
</tr>
<tr>
<td>Technical Specs. Document (TS)</td>
<td>TS_ModuleArea_ProgramName</td>
<td>TS_PY_WeeklyOvertime, which is Technical Specs document for Weekly Overtime program in the Payroll module</td>
</tr>
<tr>
<td>Functional Specs. Document (FS)</td>
<td>FS_ModuleArea_SpecsName</td>
<td>FS_MM_InventoryList which is, Functional Specs document for InventoryList in Material Management module.</td>
</tr>
<tr>
<td>SAP OSS Note document (SN)</td>
<td>SN_ModuleArea_Functionality_OSS number</td>
<td>SN_TM_overtimepay_635374/2009 which is, SAP OSS Note #635374/2009 for OvertimePay in the Time Management module.</td>
</tr>
</tbody>
</table>
15.2 SHARED DRIVE

The policies and procedures for storing, maintaining, and controlling data on LACCD’s shared drive are given below:

- **Policy** - To ensure that each department is able to efficiently access, update, and share documents and mission-critical and mission-essential data. The shared drive provides a secure location to store shared documents and data.

- All documents and data required for the day-to-day, short-term, and long-term operation of LACCD’s SAP department should be stored on the shared drive. Additionally, any document or data that is being used by two or more individuals in the SAP department may be stored in the shared drive.

- **Access** – All members in the SAP ERP team have access to the shared drive.

- **Requirements** - Shared drives are setup for each individual in the SAP department automatically. Coordinators or SAP ERP Manager may request that specific members of the SAP department be provided access to a shared drive by completing the Shared Drive Request Form.

- **Shared Drive Maintenance** - It is the responsibility of the Coordinators to stay within their allotment of disk space, ensure that all mission-critical and mission-essential documents and data are stored on the shared drive, and to archive older document versions.

- **Shared Drive Backups** - Data on the shared drive will be regularly backed up by the IT Department using the standard Server Backup procedure and timeline.
Data/Document Management - To ensure that the proper version of information is located on the shared drive, that all mission-critical and mission-essential data and documents are stored on the shared drive (and not on an employee's PC or any other place), and that employees are able to efficiently access all necessary SAP department data and documents, the following document management strategies should be followed:

- Each Coordinator must establish and follow a review process to ensure that documents/data to be placed on the shared drive are timely, accurate, and appropriate.

- A master list should be kept that tracks all documents/data stored on the shared drive.

- It is recommended that the folder structure be based upon the Module/Area, services, activities, or functions. Folders can be created for each major service, activity, or function the SAP department performs. The folder name should describe its contents (e.g. This document could be located inside the “Policies” folder.

- All documents should include a descriptive file name and data of last modification (e.g. This policy document could be named “shared_drive_policy_041309”). The file name would indicate that this is the Shared Drive Policy and it was last modified on April 13, 2009.

- Shared drive must be reviewed periodically to ensure that all appropriate documents/data are being stored on the shared drive and that the appropriate employees are able to gain access.
15. Documentation standards

Invalid or obsolete documents/data must be removed immediately when identified. Prior drafts and multiple copies of documents should not be stored on the shared drive.

Updates to documents posted on the shared drive should be approved by the SAP ERP Manager.

Folders and files will be monitored on a weekly basis and be moved to respective folders where they actually must belong.

Creation of New Folders or deletion of Files requires approval from the SAP ERP Manager.

15.3 FOLDER STRUCTURE IN SHARED DRIVE

Please click on the link to view the folder structure of the shared drive.
16. SCHEDULING BACKGROUND JOBS

16.1 INTRODUCTION

Background jobs can be defined and schedule in two ways from the Job Overview:

- Directly from transaction SM36. This is best for users already familiar with background job scheduling.
- The Job Scheduling Wizard. This is best for users unfamiliar with SAP background job scheduling. To use the Job Wizard, start from Transaction SM36, and either select Goto → Wizard version or simply use the Job Wizard button.

16.2 PROCEDURE TO SCHEDULE A BACKGROUND JOB

1. Call Transaction SM36 or choose CCMS → Jobs → Definition.

2. Assign a job name. Decide on a name for the job you are defining and enter it in the 'Job Name' field.

3. Set the job’s priority, or "Job Class":

   1. High priority: Class A
   2. Medium priority: Class B
   3. Low priority: Class C

4. In the Target server field, indicate whether to use system load balancing.

   - For the system to use system load balancing to automatically select the most efficient application server to use at the moment, leave this field empty.
   - To use a particular application server to run the job, enter a specific target server.
5. If spool requests generated by this job are to be sent to someone as email, specify the email address. Choose the Spool list recipient button.

6. Define when the job is to start by choosing Start Condition and completing the appropriate selections. If the job is to repeat, or be periodic, check the box at the bottom of this screen.

7. Define the job’s steps by choosing Step, and then specify the ABAP program, external command, or external program to be used for each step.

8. Save the fully defined job to submit it to the background processing system. When you need to modify, reschedule, or otherwise manipulate a job after you've scheduled it the first time, you'll manage jobs from the Job Overview.

Note: Release the job so that it can run. No job, even those scheduled for immediate processing, can run without first being released.

16.3 SCHEDULING JOBS WITH JOB DOCUMENTATION

To open the application for scheduling jobs from the job documentation, select a system on the Systems tab and then choose Schedule. Click on the link to get more documentation on scheduling jobs with job documentation.

16.4 BACKGROUND JOB SCHEDULING STANDARDS

- Must be scheduled by the Coordinators only
- Any change in scheduling a background job must be approved by the Manager or notified to the Manager in advance
- Any new job that is to be scheduled to run must be approved by the Manager
- Before scheduling a background job, risk analysis must be performed
- All background jobs must be documented – based on
16. Scheduling Background Jobs

- Type of background job
- Mode of background job
- Order of executing background jobs
- Monitoring background jobs
- Ad hoc background jobs

- All Background jobs must be thoroughly and constantly monitored by the Operations Group

16.5 PRODUCTION CONTROL CONSIDERATIONS

If there were no records to process, program must return the message: “No records selected”. For potentially long-running jobs, the program must produce message: “# of records processed” every 5,000 records. At the end of the processing, the program must produce the message: “# of records inserted/updated”.

16.6 COMMUNICATION BETWEEN FOREGROUND AND BACKGROUND PROCESSES

This section discusses LACCD’s recommended method of communicating information from a foreground process to a background process.

In some cases it is desirable to have an interactive program which acquires some data (perhaps by reading a file on the desktop machine, perhaps some by other means such as user entry or a selection controlled by user entry) and passes that data to a background job for further processing. There are several ways that the information could be passed. One method is to have the interactive program write a Unix file which the background job will read. That is NOT recommended (any more).
The following three methods are recommended. The first two methods assumes that the interactive (foreground) job creates and submits the background job and can pass parameters to the background job by including the parameter values in the \texttt{SUBMIT} command.

1. If there is very little data, have the foreground job use the \texttt{SUBMIT} command to pass the actual data through selection screen parameters or select-options when creating the background job.

2. If there is more data, have the foreground job export it to the \texttt{INDX} table in the database and use the \texttt{SUBMIT} command to pass the ID key through a selection screen parameter when creating the background job.

3. In some special cases, it might be better to create a new custom table and have the foreground job load the data into the custom table, from which the background job will retrieve it.

This is an expanded description of method 2, above. Assume that an internal table named "t_itab" is defined (and loaded with data) in the foreground job and the internal table "t_itab" has exactly the same definition in the background job.

In the foreground job:

\begin{verbatim}
TABLES INDX. "Not required in 4.6C
DATA key like INDX-SRTFD.
EXPORT t_itab TO DATABASE INDX (ar) ID key.
\end{verbatim}

Where "t_itab" is the name of the internal table,
"ar" is a two-character constant,
"key" is a variable containing a unique key (up to 22 characters long) that was
created (you might want to consider using elements such as the user name, the date, the time, the work process number, the application server name, etc. to produce a unique key)

SUBMIT BACKGRND ... WITH A_KEY1 = key ... .

In the background job:

TABLES INDX.
PARAMETERS: a_key1 like INDX-SRTFD NO-DISPLAY.
IMPORT t_itab FROM DATABASE INDX(ar) ID a_key1.

Where "t_itab" is the name of your internal table,
"ar" is the same two-character constant,
"a_key1" is the parameter into which the unique key was passed

After the data is no longer needed, the program should

DELETE FROM DATABASE INDX(ar) ID a_key1.
17. ABAP/4 TUNING CHECKLIST

The general performance standards below outline ways to increase efficiency from many different perspectives. The following checklist is developed by SAP to quickly review the most common performance problems.

√ - Is the program using SELECT * statements?

Convert them to SELECT column1 column2 or use projection views.

√ - Are CHECK statements for table fields embedded in a SELECT ... ENDSELECT loop?

Incorporate the CHECK statements into the WHERE clause of the SELECT statement.

√ - Do SELECTS on non-key fields use an appropriate DB index or is the table buffered?

Create an index for the table in the data dictionary or buffer tables if they are read only or read mostly. Please consult R3-Admin for creation of indexes on or the buffering of SAP supplied tables.

√ - Is the program using nested selects to retrieve data?

Convert nested selects to database views, DB jins (v4.0), or SELECT xxx FOR ALL ENTRIES IN ITAB. Click here for a discussion on "for all entries".
√ - Are there selects without WHERE condition against tables that grow constantly (BSEG, MKPF, VBAK)?

Program design is wrong - back to the drawing board.

√ - Are SELECT accesses to master data tables buffered (no duplicate accesses with the same key)?

Buffer accesses to master data tables by storing the data in an internal table and filling the table with the READ TABLE ... BINARY SEARCH method

√ - Is the program using SELECT ... APPEND ITAB ... ENDSELECT techniques to fill internal tables?

Change the processing to read the data immediately into an internal table (SELECT VBELN AUART ... INTO TABLE T_VBAK ...)

√ - Is the program using SELECT ORDER BY statements?

Data should be read into an internal table first and then sorted unless there is an appropriate index on the ORDER BY fields

√ - Is the programming doing calculations or summarizations that can be done on the database via SUM, AVG, MIN, or MAX functions of the SELECT statement?

Use the calculation capabilities of the database via SELECT SUM, ...

√ - Are internal tables processed using the READ TABLE itab WITH KEY ... BINARY SEARCH technique?
Change table accesses to use BINARY SEARCH method.

√- Is the program inserting, updating, or deleting data in dialog mode (not via an update function module)?

Make sure that the program issues COMMIT WORK statements when one or more logical units of work (luws) have been processed.
APPENDIX A : SAMPLE PROGRAMS

REPORT ZSKELREP.
*------------------------------------------------------------------------
* Purpose:
*------------------------------------------------------------------------
* Author:
* Date:
*------------------------------------------------------------------------
* Revision History:
*------------------------------------------------------------------------

TABLES: ...
DATA: ...
SELECT-OPTIONS: ...
PARAMETERS: ...
FIELD-GROUPS: HEADER,
FIELD-SYMBOLS: ...

* Event which occurs before the selection screen is shown to the user.

INITIALIZATION.

* Event which occurs each time the user hits enter on the selection screen. This event is *ignored in background processing.

AT SELECTION-SCREEN.
TOP-OF-PAGE.
END-OF-PAGE.
START-OF-SELECTION.

* Definition of fields for FIELD-GROUP extract

INSERT: ... INTO HEADER.
GET ...
END-OF-SELECTION.
SORT ...

LOOP ...
   AT FIRST.
      ENDAT.

   AT NEW ...
ENDDAT.

AT END OF ...
ENDDAT.

AT LAST.
ENDDAT.
ENDDLOOP.

*-----------------------------

*Subroutines

*-----------------------------
FORM ...
ENDFORM.

Interactive ABAP List Report

REPORT SKELINT.

*-----------------------------

* Purpose:

*-----------------------------

* Author:

* Date:

*-----------------------------

* Revision History:

*-----------------------------

TABLES: ...
DATA: ...
SELECT-OPTIONS: ...
PARAMETERS: ...
FIELD-SYMBOLS: ...

FIELD-GROUPS: ...

* Event which occurs before the selection screen is shown to the user.

INITIALIZATION.
* Event which occurs each time the user presses any button on the selection screen. This event is ignored in background processing.

```
AT SELECTION-SCREEN.
TOP-OF-PAGE.

* Top of page for sublists
TOP-OF-PAGE DURING LINE-SELECTION.
END-OF-PAGE.
START-OF-SELECTION.
GET ...
END-OF-SELECTION.

* Produce main list report SY-LSIND = 0.
SORT ...
LOOP ...
   WRITE:/ ...

* Hide specific fields which are of importance to the line
HIDE: ...
ENDLOOP.

* Event which occurs when user hits a particular F key, i.e. F6
* The hide area and SY-LISEL are automatically available.
* Produces a sublist SY-LSIND = 1-9. F3 is automatic, will always take the user back one list level, (SY-LSIND - 1).

AT PF...

* Event which occurs when a user types =LIST in the OK code
* The hide area and SY-LISEL are automatically available.
* Produces a sublist SY-LSIND = 1-9. F3 is automatic, will always take the user back one list level, (SY-LSIND - 1).

AT USER-COMMAND.
CASE SY-UCOMM.
   WHEN 'LIST'. ....
ENDCASE.
* Event which occurs when the user places the cursor on a specific line on the report and hits F2, double-clicks a line, or clicks on a hot spot.
* The hide area and SY-LISEL are automatically available.
* Produces a sublist SY-LSIND = 1-9. PF3 is automatic, will always take the user back one list level, (SY-LSIND - 1).

`AT LINE-SELECTION.`

*---------------------------------------------------------------
* Subroutines
*---------------------------------------------------------------

`FORM . . .`

`ENDFORM.`
CREATE A SEQUENTIAL DATASET

REPORT ZSKELOUT.

* Purpose:
* Author:
* Date:
* Revision History:

TABLES: ...
DATA: ...

SELECT-OPTIONS: ...

PARAMETERS: ...

FIELD-SYMBOLS: ...

START-OF-SELECTION.

GET ...

MOVE ... TO ...

WRITE ... TO ...

UNPACK ... TO ...

TRANSFER ... TO ...

END-OF-SELECTION.

* Subroutines

FORM ...
ENDFORM.
Read Sequential Dataset and Create a BDC Session

report zskelbdc.
*---------------------------------------------------------------
* Purpose:
*---------------------------------------------------------------
* Author:
* Date:
*---------------------------------------------------------------
* Revision History:
*---------------------------------------------------------------

tables: ...

data: t_bcddata like bcddata occurs 0 with header line.
data: t_messtab like bdcmsgcoll occurs 0 with header line.
data: begin of t_tab occurs ...
   end of t_tab.
select-options: ...
parameters: ...
field-symbols: ...
*---------------------------------------------------------------

start-of-selection.
   Open dataset p_dataset in text mode.
      If sy-subrc <> 0.
         Write: / text-e00, sy-subrc.
         Stop.
      Endif.
   *---- Open the BDC Session -------------------------*
   Write: /(20) 'Create group'(i01), group.
   Skip.
   Call function 'BDC_OPEN_GROUP'
      Exporting client = sy-mandt
         group = group
         user = user
         keep = keep
         holddate = holddate.
   Write: /(30) 'BDC_OPEN_GROUP'(i02), (12) 'returncode:'(i05),
          sy-subrc.
*----------- DYMPRO nnn ------------------------------------------------*
   Perform bdc_dynpro using 'sapmxxxx' 'nnn'.
   Perform bdc_field using 'TABL-FIELD' 'LITERAL'.
*----------- DYMPRO nnn ------------------------------------------------*
   Perform bdc_dynpro using 'sapmxxxx' 'nnn'.
Perform bdc_field using 'TABL-FIELD' TAB-VAR.
*---------------------------------------------------------------------*

Call function 'BDC_INSERT'
   Exporting tcode = tcode
   Tables dynprotab = t_bdcdata.
Write: /((25) 'BDC_INSERT'(i03), tcode,
      (12) 'returncode:'(i05), sy-subrc.
Close dataset p_dataset.

*----- Close the BDC Session ----------------------------------------*
Call function 'BDC_CLOSE_GROUP'.
Write: /(30) 'BDC_CLOSE_GROUP'(i04),
      (12) 'returncode:'(i05), sy-subrc.
End-of-selection.

*---------------------------------------------------------------------*
* Subroutines *
*---------------------------------------------------------------------*

*--- Add a line to the DYNPRO Table -------------------------------*

Form bdc_dynpro using program dynpro.
   Clear t_bdcdata.
   T_bdcdata-program = program.
   T_bdcdata-dynpro = dynpro.
   T_bdcdata-dynbegin = 'X'.
   Append t_bdcdata.
Endform.

*--- Add a field to a DYNPRO --------------------------------------*

Form bdc_field using fnam fval.
Clear t_bdcdata.
T_bdcdata-fnam = fnam.
T_bdcdata-fval = fval.
Append t_bdcdata.
Endform.
CALL TRANSACTION USING Technique

REPORT ZSKELCLT.

* Purpose: 
* Author: 
* Date: 
* Revision History: 

tables: indx, ...
data: return_code like sy-subr,
    Message_text(120).
data: begin of t_trantab occurs 10.
    Include structure bdcdatadata: end of trantab.

select-options: ...
parameters: ...
field-symbols: ...

start-of-selection.

*-------- DYNPRO nnn ---------------------------*
perform bdc_dynpro using 'sapmxxxx' 'nnn'.
perform bdc_field using 'TABL-FIELD' 'LITERAL'.
*-------- DYNPRO nnn ---------------------------*
perform bdc_dynpro using 'sapmxxxx' 'nnn'.
perform bdc_field using 'TABL-FIELD' TAB-VAR.
*-------- DYNPRO nnn ---------------------------*
 *
call transaction ' ' using t_trantab mode 'N' update 'S'.

* Message handling
return_code = sy-subr.
data: message_text(80) type c.
MESSAGE ID SY-MSGID
TYPE 'S' NUMBER SY-MSGNO
WITH SY-MSGV1 SY-MSGV2 SY-MSGV3 SY-MSGV4 INTO MESSAGE_TEXT.
if return_code = 0.

* At this point, various things can be done to make the process slicker.
* Send the confirmation or error to the other program via RFC.
* Store key values and confirmation document number or error message in an internal
  table for future processing.

move 'Transaction posted' TO ...
move message_text to ...
modify ...
mlse.
move 'Transaction failed' to ...
move message_text to ...
modify ...

*In the event of errored transactions:
*Store the internal table in the INDX database for future online processing of the SAP transaction.

export t_trantab to indx(..) Id ...
endif.

* Or create a batch input session for future processing.
refresh t_trantab.
end-of-selection.

*------------------------------------------------------------------
* Subroutines *
*------------------------------------------------------------------
*--- Add a line to the DYNPRO Table -------------------

form bdc_dynpro using program dynpro.
clear t_bdcdata.
t_bdcdata-program = program.
t_bdcdata-dynpro = dynpro.
t_bdcdata-dynbegin = 'X'.
append t_bdcdata.
endform.

*--- Add a field to a DYNPRO --------------------------------------

form bdc_field using fnam fval.
clear t_bdcdata.
t_bdcdata-fnam = fnam.
t_bdcdata-fval = fval.
append t_bdcdata.
endform.
SAMPLE CODE

Decision statements

Don't place decision to execute in the subroutine

```plaintext
perform get_start_end_date.

form get_start_end_date.
  check not (sword[] is initial ).
  ...more processing...
  endform.

Instead do:
if not (s_workdt[] is initial ).
perform get_start_end_date.
endif.
```

Case Statement

Use CASE statement instead of IF...ELSEIF when possible

```plaintext
if t_bsid-maber = c_alumni_debit.
  ...do something...
elseif t_bsid-maber = c_medical_debit.
  ...do something...
else.
  ...do something else...
endif.

Instead do:
Case t_bsid-maber.
  When c_alumni_debit.
    ...do something...
  When c_medical_debit.
    ...do something...
    When others.
    ...do something else...
Endcase.
```
 Nested If loops

Test most likely to fail first (specific to general)

```java
if t_covp-vrgng = 'COIN'. " a document created in FI
if t_covp-objnr = 'PR00003078'. " a specific cost object (wbs element)
    ...do something...
endif.
endif.

Instead do:
If t_covp-objnr = 'PR00003078'. " a specific cost object (wbs element 6453100)
If t_covp-vrgng = 'COIN'. " a document created in FI
    ...do something...
Endif.
Endif.
```

**And statement**

Test most likely to fail first (specific to general)

```java
if t_covp-vrgng = 'COIN' and t_covp-objnr = 'PR00003078'. "FI doc and wbs 6453100
    ...do something...
endif.

Instead do:
If t_covp-objnr = 'PR00003078' and t_covp-vrgng = 'COIN'. "6453100 and FI
    ...do something...
Endif.
```
OR Statement

Test most likely to succeed first (general to specific)

```sql
if t_covp-objnr = 'PR00003078' or t_covp-vrgng = 'COIN'. "6453100 or FI

...do something...
Endif.

Instead do:
If t_covp-vrgng = 'COIN' or t_covp-objnr = 'PR00003078'. "FI or 6453100

...do something...
Endif.
```

Sort and READ TABLE

Sort and read table t_tab with key ... Binary search when possible especially against non-buffered table (data dictionary - technical info)

```sql
loop at t_bseg.
select single name1 from kna1 into w_name
    where kunnr = t_bseg-kunnr.
endloop.

Instead do:
data: begin of t_kna1 occurs 0,
kunnr like kna1-kunnr,
name1 like kna1-name1,
end of t_kna1.
select kunnr name1 from kna1 into table t_kna1.
sort t_kna1 by kunnr.

loop at t_bseg.
read table t_kna1 with key kunnr = t_bseg-kunnr binary search.
if sy-subrc = 0.
    W_name = t_kna1-name1.
endif.
endloop.
```
Read TABLE and Modify

Read table t_vbap
With key ps_psp_pnr = t_tab-pspnr binary search.
T_vbap-kunnr = w_kunnr.
Modify table t_vbap index sy-tabix.

Modifying internal tables

When you need to modify an internal table use field symbols instead of reading table into
header and modifying.

loop at t_vbak.
select single kunnr into t_vbak-payer
from vbpa
where vbeln = t_vbak-vbeln
  and posnr = c_head_data       " 000000 item indicates header
  and parvw = c_payer.          " RG is payer
if sy-subrc <> 0.
message a056 with 'VBPA' t_vbak-vbeln.
endif.
modify t_vbak.
endloop.  " loop through contract headers and find payers

Instead do:

field-symbols <f_vbak> like t_vbak.

loop at t_vbak assigning <f_vbak>.
select single kunnr into <f_vbak>-payer
from vbpa
  where vbeln = <f_vbak>-vbeln
  and posnr = c_head_data       " 000000 item indicates header
  and parvw = c_payer.          " RG is payer
if sy-subrc <> 0.
message a056 with 'VBPA' <f_vbak>-vbeln.
endif.
endloop.  " loop through contract headers and find payers
COLLECT (building an internal table with unique key)

When you need to build a table of unique row entries and you aren't using select into collect is a useful command. It will summarize, treating all character fields as the unique key for the internal table.

If not ( iw_current is initial ).
<f_vbap>-current = <f_vbap>-current + iw_current.
Read table t_vbak key vbeln = <f_vbap>-vbeln binary search.

*by definition header must exist if line item exists! Hence no return code check

T_expenses-prime_contract = t_vbak-bstnk.  " primary contract
T_expenses-posid = <f_prps>-posid.        " billable wbs
T_expenses-matnr = w_matnr.               " material
T_expenses-kbetr = iw_current.            " current expenses
Collect t_expenses.
   Endif.

SORT by fields

SORT tables BY fields
   Sort t_kna1.
Instead,
   Sort t_kna1 by kunnr.

SELECT SINGLE

select name1 from kna1 into w_name up to 1 rows
where kunnr = t_bseg-kunnr.
Instead do:
select single name1 from kna1 into w_name
where kunnr = t_bseg-kunnr.
Or:
Select single name1 ktokk from lfa1 into (w_vname1, w_ktokk)
Where kunnr = t_bseg-kunnr.
SELECT FIELDS INTO AN INTERNAL TABLE

data: begin of t_kna1 occurs 0,
kunnr like kna1-kunnr,
namel like kna1-namel,
end of t_kna1.

select kunnr namel from kna1 into table t_kna1. "replaces contents of t_kna1

APPENDING LINES TO INTERNAL TABLE

Data: begin of t_proj occurs 0,
Pspnr like proj-pspnr, " internal project def. Number
Objnr like proj-objnr, " PD + internal proj number
End of t_proj.

Data: begin of t_prps occurs 0,
Objnr like prps-objnr, " WBS element object # PR+int rsn
Pspnr like prps-pspnr, " WBS element internal rsn
Posid like prps-posid, " WBS element external number
Stufe like prps-stufe, " hierarchy level (1 is top)
Fakkz like prps-fakkz, " billing indicator (X=yes)
Zbillt like prps-zbillt," billing type (01=cost reimburs)
Psphi like prps-psphi, " number of project def. Internal
Zend like prps-zend, " expiration date
Ztermcd like prps-ztermcd," term code
Up like prhi-up, " parent WBS element internal
Budget like rpsco-wtp00, " auth. Total on WBS element
Actual like rpsco-wtp00, " total expenditure on WBS
Revenue like rpsco-wtp00, " total revenue on WBS
Begbal like rpsco-wtp00, " begin balance on WBS
End of t_prps.

Refresh t_prps.
Clear t_prps.

Loop at t_proj.
  Select objnr pspnr posid stufe fakkz zbillt psphi zend ztermcd
  From prps into t_prps "place row into header of t_prps
  Where psphi eq t_proj-pspnr "prps project = project rsn
  And zbillt eq p_billt. "billing type input param (01)
  Append t_prps."append header to body of table
  Endselect. "notice no keyword table which is why
  the endselect

If sy-subrc <> 0.
  Write t_proj-pspnr to w_posid. "convert to wbs element ext #
W_error = 'Project & with no WBS element'(002).
Replace '&' with w_posid(7) into w_error.
Write w_error.
Endif.
Endloop. "loop through all the billable projects

For all entries faster than appending

Instead of appending (as in above example) do the following (but make sure unique key is in select clause):

Select objnr pspnr posid
  stufe fakkz zbillt
  psphi zend ztermcd "pspnr is unique
From prps into table t_prps "place row into header of t_prps
For all entries in t_proj "all billable projects in internal table
  Where psphi eq t_proj-pspnr "prps project = project rsn
  And zbillt eq p_billt. "billing type input param (01)

If the select clause is not unique, it is the equivalent of using a "select distinct". In other words duplicates will not be selected.

Corresponding fields

Data: begin of t_kna1 occurs 0,
  Kunnr like kna1-kunnr,
  Name1 like kna1-name1,
End of t_kna1.

Select name1 from kna1 into corresponding fields of table t_kna1.

"For all entries in" - 3 pitfalls to avoid

Select shkzg wrbtr saknr "debit/credit indicator, amount, GL acct
From bseg into table t_bseg
For all entries in t_bkpf
  Where belnr = t_bkpf-belnr and bukrs = 'CUR '.
1) This is the equivalent of saying "select distinct shkzg wrbtr saknr"

It will only pick up one line item if multiple line items appear with the same debit/credit indicator, amount and GL Account. If you want all occurrences of these you must have a select statement that includes the table’s unique key, also called primary key. In the Data dictionary (SE11) the fields belonging to the unique key are marked with An "X" in the key column.

Instead do:

```
Select bukrs belnr gjahr buzei shkzg wrbtr saknr "bseg unique key + d/c ind, amt, GL acct
From bseg into table t_bseg
For all entries in t_bkpf
Where belnr = t_bkpf-belnr and bukrs = 'CUR '.
```

2) FOR ALL ENTRIES IN...acts like a range table, so that if the "one" table is empty, all rows in the "many" table are selected.

```
If not t_proj[] is initial.
Select objnr pspnr posid stufe fakkz zbillt psphi zend ztermcd
From prps into table t_prps
For all entries in t_proj
    "all billable projects
Where psphi eq t_proj-pspnr
And zbillt eq p_billt.
    "billing type input param (01)
Endif.
    "if there are any projects
```

3) If the parent table (t_proj) is very large there is performance degradation Instead loop through parent table(t_proj), appending to child table (t_prps)

```
Loop at t_proj.
    Select objnr pspnr posid stufe fakkz zbillt psphi zend ztermcd
        From prps into t_prps
        "place row into header of t_prps
```
Where psphi eq t_proj-pspnr
And zbillt eq p_billt.
Append t_prps.
Endselect.

If sy-subrc <> 0.
Write t_proj-pspnr to w_posid.
W_error = 'Project & with no WBS element'(002).
Replace '&' with w_posid(7) into w_error.
Write w_error.
Endif.
Endloop.

Where clause (rules based) example

where objnr = c_offset
and lednr = '00'
and versn = '000'.
and wrttt = c_actual
and gjahr =< gjahr
and kstar = c_overrun
and beknz <> c_settlement.

Delete all rows from table

Delete from zzsupradr client specified where mandt = sy-mandt.

Append one table to another when structures are the same

Data t_name like is_name occurs 0 with header line.
Data t_prob like is_name occurs 0 with header line.

loop at t_name where room = space.
write: / t_name-name, t_name-room, t_name-cnt.
move t_name-name to t_prob-name.
move t_name-room to t_prob-room.
move t_name-cnt to t_prob-cnt.
append t_prob.
endloop.

Instead do:

loop at t_name where room = space.
write: / t_name-name, t_name-room, t_name-cnt.
append t_name to t_prob.
endloop.

Check statements versus where clause criteria

Loop at t_prps where tabid = w_posid. "problem wbs elements
   Select distinct meinb kstar into table t_coep "collect unit meas.
   From coep "& cost elements from trans
   Where lednr = '00' "encourage index coep_1
   And objnr = t_prps-objnr "wbs object #
   And wrttp in r_wrttp "31,03-12 actual value types
   And versn = '00' "encourage index coep_1
   And kstar in r_kstar_cosp. "expense cost elements

   Check sy-subrc = 0. "expenses found for this wbs
element continue otherwise get next wbs

   Loop at t_coep.
      Check t_coep-meinb = w_meinb. "problem unit of measurement
   ...
   Endloop. "loop through cost elements and units of measurement
Endloop. "loop through wbs elements with potential problem

Instead do:

Loop at t_prps where tabid = w_posid. "problem wbs elements
   Select distinct kstar into table t_kstar "collect potential
From coep

Where lednr = '00'
And objnr = t_prps-objnr
And wrttp in r_wrttp
And versn = '000'
And kstar in r_kstar_cosp
And meinb = w_meinb.

Check sy-subrc = 0.

...  
Endloop.

Use text elements or constants

Use text elements for translatable text and use constants for non-translatable literals. This sample code is a good example because it uses the text element (012) in conjunction with the hard coded text. This documents the text element and provides for the possibility of multi-language support.

**Constants:**

C_csks(4) value 'CSKS',  " Cost centers
Write: / c_csks, '--- # of rows need be updated: '(012), w_cnt
SQL STATISTICAL FUNCTIONS

*-----------------------------------------------------------------------------------------------*
* CALCULATE AUTHORIZED TOTAL PER WBS ELEMENT                                               *
*-----------------------------------------------------------------------------------------------*

Select wtp00 from rpsco into table t_rpsco   "summarized budget
Where objnr = t_prps-objnr                   "PR+rsn wbs elem object
And wrttp eq c_budget.                      "41 budget value type
Loop at t_rpsco.
   Add t_rpsco-wtp00 to t_prps-budget.  "total budget
Endloop.

Instead do:

Select sum( wtp00 ) from rpsco into t_prps-budget  "summarized budget
    Where objnr = t_prps-objnr              "PR+rsn wbs elem object
    And wrttp eq c_budget.                 "41 budget value type

FORM Parmeter

Perform status_change using c_status.
Form status_change using value(p_c_status).
Endform.

Test return codes or handle the possibility that the unexpected happens

Example 1:

Loop at t_bseg.
   Clear w_name.
   Read table t_knal with key kunnr = t_bseg-kunnr binary search.
   If sy-subrc = 0.
      W_name = t_knal-name1.
   Endif.
Endloop.
Example 2:

```sql
loop at t_proj.
select objnr pspnr posid stufe fakkz zbillt psphi zend ztermcd
from prps into t_prps
  place row into header of t_prps
where psphi eq t_proj-pspnr
and zbillt eq p_billt.
append t_prps.
endselect.

If sy-subrc <> 0.
  Write t_proj-pspnr to w_posid.
  convert to wbs element ext #
    W_error = 'Project & with no WBS element'(002).
  Replace '&' with w_posid(7) into w_error.
  Write w_error.
Endif.

see below for explanation of why the return code is tested after endselect
Endloop.

loop through all the billable projects

Example 3:

```sql
select vbeln
  uses VBAK______A index
from vbak into table t_vbak
where vbeln in r_vbeln
  billing request doc range '0070000000' - '0074999999'
    and erdat = max_datum
    and auart = c_zpsr
    and vkorg = c_vkorg
    and ernam = t_zjobrun-uname.
    original run date
    billing request ZPSR
    sales org 1000 (sponsored programs)
    user who ran original job

If sy-subrc <> 0.
  Clear w_samedate.
  no bills were created
Exit.
Endif.
```
Example 4:

Call transaction 'KB14'
Using t_bdctab mode 'N' messages into t_msgtab.
If sy-subrc = 0.
Write: / t_doc-posid(8),
    'Overexpenditure posting'(038), t_doc-doc01,
    'Reversed:'(039), sy-msgvl(10).
    Commit work.
Else.
    Reversal_trouble = 'X'.
    Clear t_error.
    T_error-posid = t_doc-posid.
    T_error-vbeln = t_doc-doc01.
    Perform process_errors.
    T_error-sortkey = c_error_reversed.
    Append t_error.
    Perform bdc_session_create using 'Reverse' 'KB14'.
Endif.  "was the call transaction successful

Example 5:

(1) If a function module has no exceptions, sy-subrc will always be 0. An example of such a
function module is Z_R3_LIST_REPORT_SELECTIONS. It is pointless to test a return code
from such a function module and the "pattern" button doesn't suggest it.

(2) If a function module always uses the construct, MESSAGE ... RAISING ..., then it is
handling exceptions with its own error messages. Unless you want processing to continue
even if an error is encountered, you can drop the "EXCEPTIONS" section when you call the
function module. For instance, below is an example of redundant code created using the
"pattern" button.
Call function 'SSF_FUNCTION_MODULE_NAME'
exporting
   formname = p_form
importing
   fm_name = w_fmname  " generated function module
exceptions
   no_form = 1
   no_function_module = 2
   others = 3.
   If sy-subrc <> 0.
      Message id sy-msgid type sy-msgty number sy-msgno with sy-
      msgv1 sy-msgv2 sy-msgv3 sy-msgv4.
Endif

This code should be changed to:
call function 'SSF_FUNCTION_MODULE_NAME'
exporting
   formname = p_form
importing
   fm_name = w_fmname.  " generated function module
* function module is handling bad return code with appropriate error messages

The comment is helpful for the reviewer but should not be mandatory.

(3) If a function module uses the construct, RAISE ..., then it short dumps when an
exception is encountered! Regardless of whether you want to continue or stop processing,
you need to handle bad return codes from such function modules to avoid a short dump.
The following code would cause a short dump in the unlikely event that p_domain is
invalid:
call function 'DDIF_DOMA_GET'
exporting
    name          = p_domain
    langu         = sy-langu
    tables
        dd07v_tab     = t_dd07v.

On the other hand the following code, produced by the "pattern" button, won't work because the system fields sy-msgid, sy-msgty, sy-msgno etc, are not filled when the construct, RAISE ...., is used:

call function 'DDIF_DOMA_GET'
    exporting
        name          = p_domain
        langu         = sy-langu
        tables
            dd07v_tab     = t_dd07v
    exceptions
        illegal_input = 1
        others        = 2.
If sy-subrc <> 0.
    Message id sy-msgid type sy-msgty number sy-msgno
        with sy-msgv1 sy-msgv2 sy-msgv3 sy-msgv4.
Endif.

There are many valid ways to handle return code checking on a function module like this. Three reasonable examples are:

1. call function 'DDIF_DOMA_GET'
    exporting
        name          = p_domain
        langu         = sy-langu


tables
  dd07v_tab = t_dd07v
exceptions
  others = 1.
If sy-subrc <> 0.
  P_desc = p_code.
  Exit.
  Endif.

2. Call function 'DDIF_DOMA_GET'
   exporting
     name = p_domain
     langu = sy-langu
tables
  dd07v_tab = t_dd07v
exceptions
  others = 1.

* sy-subrc  will always be 0 because p_domain is always valid
The comment is helpful for the reviewer but should not be mandatory.

3. Call function 'JOB_CLOSE'
   exporting
     jobname = my_job-jobname
     jobcount = my_job-jobcount
     strtrimmed = c_yes
exceptions
  cant_start_immediate = 1
  invalid_startdate = 2
  jobname_missing = 3
job_close_failed = 4
job_nosteps = 5
job_notex = 6
lock_failed = 7.

If sy-subrc eq 0. 
   "Job & number & started in background message i180(26) with my_job-jobname my_job-jobcount.
Else.
   Case sy-subrc.
      When 1.
         W_problem = 'Can not start immediate'(071).
      When 2.
         W_problem = 'Invalid start date'(070).
      When 3.
         W_problem = 'Jobname missing'(069).
      When 4.
         W_problem = 'Job close failed'(068).
      When 5.
         W_problem = 'Job no steps'(013).
      When 6.
         W_problem = 'Job notex'(010).
      When 7.
         W_problem = 'Lock failed'(007).
      When others.
         W_problem = sy-subrc.
   Endcase.
* System error: Termination in routine JOB & &
   message e040(zz) with 'CLOSE'(014) w_problem.
Endif.

Example 6:
If w_batch < 999.
   Lw_action = 'insert'(041).
   Zjobrun-mandt = sy-mandt.
   Zjobrun-pgnam = w_report_name.
   W_batch = w_batch + 1.
   Zjobrun-batch = w_batch.
   Zjobrun-uname = sy-uname.
   Insert into zjobrun values zjobrun.
Else.
   Lw_action = 'update'(042).
   Update zjobrun
      set datum = lw_date
      uzeit = lw_time
      uname = sy-uname
      reccount = 0
      errcount = 0
      cntrlrec = 0
      credit = 0
      debit = 0
      errfile = ' '
      review = ' '
      text = ' '
      where pgnam = w_report_name
      and batch = w_batch.
Endif.

If sy-subrc = 0.
   Commit work.
Else.
   * Required update of table ZJOBRUN failed
   message i032 with lw_action sy-subrc.
   Stop.
Endif.
Example 7: (dataset)

Open dataset w_fname for input in text mode " get input data
message lw_msg. " any errors place here
if sy-subrc <> 0.
   Message i002 with lw_msg.
   Stop.
Endif.

Read dataset w_fname into lw_row.
   If sy-subrc <> 0. " end of file reached
      exit " leave do loop
endif.

Loop at lt_files.
Delete dataset lt_files-name.
If sy-subrc = 0.
   T_body-line = lt_files-name.
   Append t_body.
   Endif.
Endloop.

Form write_unix_file.
Open dataset myfile for output in text mode.
If sy-subrc <> 0.
   Write sy-subrc to w_subrc.
   Concatenate 'Problem opening' myfile 'to UNIX.' w_subrc
   into t_body-line separated by space.
   Append t_body.
   Concatenate p_maber 'Invoice Load problems encountered'
   into w_subject.
   Perform send_mail.
   Message a000 with 'Problem opening UNIX.
   sy-subrc." stop processing because transfer will short dump
endif. " problem with open
loop at t_invoice.
Transfer t_invoice to myfile.
Endloop. "loop through invoices
close dataset myfile.

Handling Dates

Example 1: Dates being used in a BDC session or Call Transaction

Data:

W_bldat(10), "doc date - string conversion
W_budat(10), "posting date - string conversion
D_bldat like sy-datum, "doc date - today's day
D_budat like sy-datum. "posting date - last day of month
D_bldat = sy-datum. "doc date
Call function 'RP_LAST_DAY_OF_MONTHS'

Exporting
Day_in = d_bldat

Importing
Last_day_of_month = d_budat

Exceptions
Day_in_no_date = 1.
If sy-subrc = 0.
Write d_bldat to w_bldat. "WRITE uses conversion exit to format date
Write d_budat to w_budat. "allows for correct user defaults
Endif.

Perform dynpro using:
'X' 'SAPMF05A' '0100,'
'BKPF-BLDAT' w_bldat, "07/09/1999 if user's defaults are mm/dd/yyyy
' ' 'BKPF-BUDAT' w_budat, "07/31/1999
' ' 'BKPF-BLART' w_blart, "doc type from zardiv
' ' 'BKPF-BUKRS' w_bukrs, "company code CUR
Example 2: Default dates being set during INITIALIZATION

```
data w_date like sy-datum. "used to build defaults
parameters: p_start like bsid-zfbdt default '', "begin baseline date
               p_end   like bsid-zfbdt default ''. "end baseline date
initiation.
w_date = sy-datum.
w_date+6(2) = '01'.
w_date = w_date - 1.
w_tdate = w_date.
w_date+6(2) = '01'.
w_fdate = w_date.
if sy-batch eq ' '.
p_end = w_tdate. "do not WRITE w_tdate to p_end!
p_start = w_fdate. "19990601 actual 06/01/1999 displays 1/19/06/0 if
write had been used
endif.
```

Example 3: Dates being used in the SUBMIT statement

```
constants:
c95(2) type c value '95', " 95 contract
C1995(5) type c value 'F1995', " 1995 prime contract
C20(3) type c value 'F20'. " contracts in 21st cent
perform find_prime_contract using <f_vbrk>-zuonr+7(2)
     changing <f_vbrk>-contract.
```
Form find_prime_contract using p_year type c "note that you can't specify length
Changing p_contract like vbak-vbeln.
If p_year = c95. " '95'
P_contract = c1995. " 'F1995'
Else.
Concatenate c20 p_year into p_contract. " F2000, F2005...
Endif.
Endform. " find_prime_contract

Complex Subqueries

1. To find all the personnel areas/subareas belonging to campus

   select werks btrtl btext into table t_t001p
   from t001p
   where werks like 'C%'.

   Loop at t_t001p. " loop through campus personnel areas/subareas

2. To select all the open positions for each personnel area/subarea the position must belong to an organization but must not have any people assigned to the position

   Select objid persa btrtl into table t_hrp1008
   from hrp1008 as hl
   where plvar = '01' " current plan
   and otype = 'S' " positions
   and subty = ' ' " subtype
   and istat = '1' " active
   and begda <= sy-datum " beginning on or before now
   and endda >= sy-datum " ending on or after today
   and persa = t_t001p-werks " personnel area
   and btrtl = t_t001p-btrtl " personnel subarea
   " belonging to org currnt plan
and exists ( select * from hrp1001 as h2 where plvar = '01'
and otype = 'S' and h2~objid = h1~objid
and rsign = 'A' and relat = '003' " belongs
and istat = '1' " active
and sclas = 'O' )

organization and not exists ( select *
from hrp1001 as h3
where plvar = '01'
and otype = 'S'
and h3~objid = h1~objid
and rsign = 'A' and relat = '008' " held by
and istat = '1' " active
and sclas = 'P' ).

clear w_subtotal.

Loop at t_hrp1008.

W_subtotal = w_subtotal + 1.
W_total = w_total + 1.

Write: / t_hrp1008~objid, t_hrp1008~persa, t_hrp1008~btrtl,
t_t001p~btext.

Endloop.

Write: / 'Total available for', t_t001p~btext, w_subtotal.

Uline.

Skip 2.

Endloop.

Write: / 'Grand Total available:', w_total.
APPENDIX B: TWO/THREE-CHARACTER CODES FOR APPLICATION AREAS

The following two-character abbreviations are used to identify certain projects or modules when naming programming objects. (Note that SAP has associated many two-character codes with specific application areas, even though they do not appear in this list.)

Please do not use any code not listed here, and do not use a code for any application area other than the one associated with the code in this list.

If you have an application area which is not listed here, please discuss the situation with Coordinators.
### Appendix B: Two/Three-Character Codes for Application Areas

<table>
<thead>
<tr>
<th>2/3-Character Abbr.</th>
<th>Application Name</th>
<th>Devel. Class</th>
<th>Devel. Class</th>
</tr>
</thead>
<tbody>
<tr>
<td>AM</td>
<td>Asset Management</td>
<td>ZKG0</td>
<td>.</td>
</tr>
<tr>
<td>AP</td>
<td>Accounts Payable</td>
<td>.</td>
<td>.</td>
</tr>
<tr>
<td>BN</td>
<td>Benefits</td>
<td>.</td>
<td>.</td>
</tr>
<tr>
<td>BPS</td>
<td>ZAR0</td>
<td>.</td>
<td>.</td>
</tr>
<tr>
<td>BW</td>
<td>Business Warehouse</td>
<td>.</td>
<td>.</td>
</tr>
<tr>
<td>CE</td>
<td></td>
<td>.</td>
<td>.</td>
</tr>
<tr>
<td>CO</td>
<td>Controlling</td>
<td>ZKG0</td>
<td>.</td>
</tr>
<tr>
<td>DSM</td>
<td>Data Sync Manager</td>
<td>.</td>
<td>.</td>
</tr>
<tr>
<td>EP</td>
<td>Internet Portal</td>
<td>ZEP0</td>
<td>.</td>
</tr>
<tr>
<td>FI</td>
<td>Financial Accounting</td>
<td>ZFI0</td>
<td>.</td>
</tr>
<tr>
<td>GM</td>
<td>Grants Management</td>
<td>ZGM0</td>
<td>.</td>
</tr>
<tr>
<td>HR</td>
<td>Human Resources</td>
<td>ZHR0</td>
<td>.</td>
</tr>
<tr>
<td>MM</td>
<td>Material Management</td>
<td>.</td>
<td>.</td>
</tr>
<tr>
<td>OM</td>
<td>Organizational Management</td>
<td>.</td>
<td>.</td>
</tr>
<tr>
<td>PA</td>
<td>Personnel Administration</td>
<td>.</td>
<td>.</td>
</tr>
<tr>
<td>PBC</td>
<td></td>
<td>.</td>
<td>.</td>
</tr>
<tr>
<td>PAT</td>
<td></td>
<td>.</td>
<td>.</td>
</tr>
<tr>
<td>PCR/PCS</td>
<td>Personal Change Request</td>
<td>.</td>
<td>.</td>
</tr>
<tr>
<td>PD</td>
<td></td>
<td>.</td>
<td>.</td>
</tr>
<tr>
<td>PR</td>
<td>Protocol</td>
<td>.</td>
<td>.</td>
</tr>
<tr>
<td>PY</td>
<td>Payroll</td>
<td>ZPY0</td>
<td>.</td>
</tr>
<tr>
<td>PRT</td>
<td>Payroll Reckon Tool</td>
<td>.</td>
<td>.</td>
</tr>
<tr>
<td>R3</td>
<td>LACCD Basis</td>
<td>ZR30</td>
<td>.</td>
</tr>
<tr>
<td>PS</td>
<td></td>
<td>.</td>
<td>.</td>
</tr>
<tr>
<td>TM</td>
<td>Time Management</td>
<td>.</td>
<td>.</td>
</tr>
<tr>
<td>WF</td>
<td>Workflow</td>
<td>ZWF0</td>
<td>.</td>
</tr>
</tbody>
</table>
APPENDIX C: DEVELOPMENT CLASSES

SAP Repository objects are classified according to development classes. It is recommended that functionally related objects (such as programs, dictionary components, message classes, etc...) be grouped together by development class. For example, code development for Accounts Payable would be in development class ZAP0. This structuring of the SAP object repository facilitates the control and recording of development efforts.

<table>
<thead>
<tr>
<th>Development Class</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Z001</td>
<td>Customer Development Class</td>
</tr>
<tr>
<td>ZAP0</td>
<td>Accounts Payable</td>
</tr>
<tr>
<td>ZAR0</td>
<td>Accounts Receivable</td>
</tr>
<tr>
<td>ZBC404</td>
<td>ABAP Object Class</td>
</tr>
<tr>
<td>ZEP0</td>
<td>Internet Portal</td>
</tr>
<tr>
<td>ZHR0</td>
<td>H/R Application: Customer Development</td>
</tr>
<tr>
<td>ZKG0</td>
<td>Controlling</td>
</tr>
<tr>
<td>ZPY0</td>
<td>Payroll Accounting (Employee and Pension)</td>
</tr>
<tr>
<td>ZR30</td>
<td>R/3 Administration</td>
</tr>
<tr>
<td>ZUT0</td>
<td>Tools &amp; Utilities</td>
</tr>
<tr>
<td>ZWB0</td>
<td>SAP Workbench Training: Instructor</td>
</tr>
<tr>
<td>ZWB1</td>
<td>SAP Workbench Training: Workshop</td>
</tr>
<tr>
<td>ZWF0</td>
<td>SAP Business Workflow</td>
</tr>
</tbody>
</table>

Notes: The development class Z001 is an SAP example and it should not be used to classify LACCD objects.

The Development classes ZBR1, ZGN0, and ZSPO are obsolete and should not be used to classify any new development object.
The Development classes ZWB0 and ZWB1 are meant to be specific to objects related to LACCD’s ABAP workshops. Such objects should not be transported into our test, production, or training environments.
APPENDIX D: LACCD REPORT SPECIFICATION FORM

Laccd report specification form can be found at
\\down025\grpshare\SAP\Policies\Templates\LACCD Report Specification Form v3.doc.

To help us meet your reporting needs, please provide specific details of the business process as well as your reporting requirements. Complete the Report Specification Form and send it to coroneee@email.laccd.edu

Instructions and Process

- All requests must be documented using the LACCD Report Specification Form below.
- The completion of this form requires the functional and technical staff working together. The functional user should be able to complete questions 1 through 5 and then work with the technical support staff to complete questions 6 through 8.
- A Reports Project Log will be maintained where a working committee will assign priorities, resources, etc.
- Either the functional or technical resource must report progress to the working committee.
ABOUT YOURSELF

<table>
<thead>
<tr>
<th>Requestor’s Name</th>
<th>Phone</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Dept. or Cost Center</th>
<th>Email</th>
<th>@email.laccd.edu</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Report User (or Dept.) if different from requestor</th>
<th>Todays’ Date</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>

ABOUT THE REPORT

Priority or need for this report (check one): □ High □ Medium □ Low

NAME OF DESIRED REPORT

Give a short description of the desired report (including the business benefit of this report):

Source of business data: Name the application or data area of the report (e.g., FM-Funds Management,
Please list any standard SAP or DEC reports you found that are similar to the desired report. Specify why the reports you considered do not meet your needs.

<table>
<thead>
<tr>
<th>Title of Report</th>
<th>Program name / transaction code / other</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Please enter a *long* description of the desired report. Please include:

1. The business need for this report
2. A description of the layout and screen information the report should provide (such as weekly, monthly, or quarterly data; column or row headings; summations, and document data, etc.)
3. References to any legacy system reports considered from which this report can be found (with attachments, listings, or screen-shots if possible)

<table>
<thead>
<tr>
<th>Provide detail information on columns, sorting, selection parameters, totals, subtotals, page breaks, etc. (A worksheet with a prototype of the report would provide the best example)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Include a report sample layout.</td>
</tr>
</tbody>
</table>

| Specify any known tables and/or fields desired within the report: |
Related to research done on reporting tools, specify:

1. Which reporting tools you have considered could be used to create the report you desire
2. Which reporting tool you might prefer for development of this report
3. Why certain reporting tools are inappropriate for your needs on this report
4. Feel free to use the table that follows and/or the open area below it.

<table>
<thead>
<tr>
<th>Reporting Tool or Data Collector</th>
<th>Considered?</th>
<th>Preferred for Development?</th>
<th>Explain Why Preferred or Not?</th>
</tr>
</thead>
<tbody>
<tr>
<td>ABAP Query</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Logistics Info System (LIS)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Report Painter or Report</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Writer</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>---</td>
<td>---</td>
<td>---</td>
<td></td>
</tr>
<tr>
<td>Drill-Down Reporting</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>DEC report</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Others?</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Assignments and Due Date</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Person responsible – IT</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Person responsible – Functional</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Person responsible – BW</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Due Date:</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Log Reference number:</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
APPENDIX E: PROJECT PLAN TEMPLATE

A Project Plan template can be found at \down025\grpshare\SAP\Policies\Templates\Project Plan Template.doc.
APPENDIX F: TECHNICAL SPECIFICATIONS
TEMPLATE

The Technical Specifications template can be found at
\down025\grpshare\SAP\Policies\Templates\LACCDTech_Spec_temp_v1.doc.