**NATIONAL ARCHIVES**

**AND**

**RECORDS ADMINISTRATION**

**(NARA)**

**Release Management Plan**

**(RMP)**

Version: 2.0

Status: Final

Date: 02/08/23

**Document Change Control Sheet**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Date** | **Version** | **Author** | **Status** | **Revision Description** |
| 05/05/22 | 1.0 | Seema Dheman | Final | Final document |
| 02/01/23 | 2.0 | Michael Punch/ Seema Dheman | Final | Added Naming and Labeling Guide to document |

**Table of Contents**

Contents

[1.0 Introduction 3](#_Toc126160553)

[1.1 Purpose 3](#_Toc126160554)

[1.2 Scope 3](#_Toc126160555)

[2.0 Planning the Release 4](#_Toc126160556)

[2.1 Building the Release 4](#_Toc126160557)

[2.2 Testing of the Release 4](#_Toc126160558)

[2.3 Deploying the Release 4](#_Toc126160559)

[2.4 Post Deployment Test Results 4](#_Toc126160560)

[2.5 Capture Deployment Artifacts 4](#_Toc126160561)

[2.6 Reference Documents 5](#_Toc126160562)

[3.0 Release Management Naming and labeling Guide 5](#_Toc126160563)

[3.1 Assigning Release Labels and Labeling Software Code 5](#_Toc126160564)

[3.2 Labeling Electronic Customer Relationship Management (ECRM) Releases 6](#_Toc126160565)

[4.0 Types of Releases 7](#_Toc126160566)

[4.1 Release and Deployment Process 7](#_Toc126160567)

[4.2 Release Management Checklist 7](#_Toc126160568)

# 1.0 Introduction

This Release Management Plan (RMP) seeks to ensure a balanced and logical approach to managing software releases across all Projects within Information Services (I). More specifically, this Plan addresses dependencies between many of the critical areas, both direct and indirect, in the software release and deployment maintenance lifecycle.

The strategy of Release Management (RM) is to plan into the future; thereby allowing for sufficient development and testing time for ongoing production support and future initiatives while addressing customer needs. The RMP alignment of distinct functional and technical processes lends itself to evolving improvements in the efficiency of deployments that in turn promote an environment where organizational priorities become better managed, and resources better utilized.

The overarching goal of Release Management (RM) is to protect the integrity of the production environment, manage the risks associated with the deployment of a release and ensure all changes are addressed as efficiently as possible thereby reducing deployment costs and increasing customer satisfaction. Further, it lays the foundation for a repeatable process supporting operational stability and predictability.

## 1.1 Purpose

The purpose of this RMP is to describe the approach for managing releases and their eventual deployment into a Production environment. This plan provides a framework for the execution of release management in an organized and logical fashion. It also contributes to the effective use of Information Services (I) resources in building, managing, and delivering code in order to satisfy customer needs.

## 1.2 Scope

This plan gives a high-level and brief description of the many touch points that play key roles in the RM function. It also sets forth the goals and objectives of our release management efforts, defines the various types of releases, and addresses roles and responsibilities. The high-level components that comprise an effective RMP strategy are:

* Planning the release
* Building the release
* Testing the release
* Deploying the release
* Post deployment test results
* Capture deployment artifacts

# 2.0 Planning the Release

Release planning entails engaging all the affected parties, such as the vendor, testing, security, Project Manager, and the customer, to come to an agreement on a schedule and release content that works for all the stakeholders. Identifying resources, communicating to stakeholders, and capturing test results are all part of release planning. The Release Manager takes the following criteria into account when planning a release:

* The type of release being proposed – major, minor, or emergency.
* Appropriate change control board approval – Project CCB or ECAB.
* Resources available for testing, distributing, implementing, and communicating the change.
* Identification of impacts and dependencies to deployments already scheduled.

## 2.1 Building the Release

The Release Manager ensures that the release has successfully passed IQ testing and any issues are resolved before the release is promoted to the production environment.

## 2.2 Testing of the Release

The RM verifies that IQ Test Team, or in some cases the vendor for releases that change hardware configurations and don’t involve an actual code release, provides a “Go” recommendation for the release. All releases, without exception, must pass testing before being deployed.

## 2.3 Deploying the Release

The RM ensures the Request For Change (RFC) was approved and issues an Authorization to Deploy email to all interested parties. The RM also ensures that a system maintenance banner is posted prior to deployment, smoke testing activities are identified and agreed to prior to RFC approval, and that a roll back plan is in place.

## 2.4 Post Deployment Test Results

The Release Manager verifies with the appropriate parties (Users, Ops team, Vendor, PM) that the deployment was successful and release the post deployment testing results to Configuration Management in order to close the RFC.

## 2.5 Capture Deployment Artifacts

The release is documented (artifacts collected) and any known defects and workarounds are identified. All source files are provided to the CM Library by the Release Manager after obtaining system engineering’s approval that all the appropriate files are present and accounted for. The Vendor or the Release Manager prepares a Version Description Document (VDD) for every release that entails a code change. Configuration changes do not require a VDD be prepared. Deployment artifacts generally consist of affected documents, the source and object code, post deployment smoke test results, and any other items that support RFC closure. w

## 2.6 Reference Documents

* Project Configuration Management Plan (CMP)
* QA Naming and Labeling Guide
* Release Management Checklist
* Release Management Process Workflow Diagram
* Release Management Calendar(s)

# 3.0 Release Management Naming and labeling Guide

### 3.1 Assigning Release Labels and Labeling Software Code

The following describes the labeling convention for software releases:

**Project(s) with Subproject(s)**

**Project\_SubProject\_XX.Y.Z.Za**

Examples: **ERA**\_**Base**\_**12**.**1.1.0a**

**ERA2.0\_ADIS\_18.1.1.1**

**Where:**

Project = name of the Project e.g. ERA, ERA 2.0, EOP etc.

Subproject = is one of the following: AAD, BASE, ADIS, NAC, DAS, CRI, EOP43, EOP44 OPA for ERA and PERL, ADDRESS and URTS for NDC etc.

XX = the last two digits of the current calendar year.

Y = the Release Number, e.g. 1, 2, 3…, …10, 11, 12, etc.

This number is assigned in incrementing order and corresponds to new major developments. (Major fixes)

Z.Z = is the iterative Build Number, e.g. 0.0, 1.0, 2.0 (major build)

and / or 1.1. 1.2. (minor build)

a- denotes an emergency change

Note: A Version Description Document (VDD) is not required for "configuration file”, “scripts”, or “database” type deployments. These types of files are not normally under CM control but are delivered to CM as required for safekeeping.

**Full Builds will be delivered in the months of January and June for NAC and DAS only**. A full build is denoted by the release number ending with a “zero”, i.e., NAC\_18.1.1.0. A partial build example would be NAC\_18.1.1.1, or NAC 18.1.1.2. Other projects will provide Full Builds with every release.

A partial build consists of only the files that were changed for that build (usually a subset of the release) whereas a major build consists of all the files necessary to reconstitute the application.

**Note: If there are no sub-projects then it will be Project name and the release number. Sub projects will not be required.**

1. **Labeling Commercial off-the-shelf (COTS) Releases**

The COTS system releases use the following format:

**PRJ\_yy.mm.wn**

PRJ\_= COTS

yy = Two (2) -digit year number (e.g., 09, 10, 11) followed by a period

mm = Two (2)-digit month number (e.g., 01-12) followed by a period

wn = Single-digit week number (e.g., 1-5)

Example: COTS\_15.01.2 = A planned COTS SW Release for week 2 of January 2015

COTS\_15.05.5 = A planned COTS SW Release for week 5 of May 2015

## 3.2 Labeling Electronic Customer Relationship Management (ECRM) Releases

ECRM COTS SW releases will use the following labeling convention:

“PRJ\_ECRM Seasonal Release Number\_System Instance\_YY.BB”

PRJ = COTS SW platform which is ECRM (Salesforce)

ECRM Seasonal Release Number = is generated/assigned by the ECMR (Salesforce) platform and changes three times a year: SPRyy, SUMyy, and WINyy, where SPR is Spring, SUM is Summer, and WIN is Winter and “yy” is the year.

System Instance = System/Application being changed, i.e. RRS, ETMS, etc.

YY = Two (2) -digit year number (e.g., 18, 19, 20) followed by a period

BB = Two (2) -digit application build number (e.g.1.0, 2.0 and so on)

Example: ECRM\_SPR20\_RRS\_20.1.0 where ECRM is the COTS S/W platform, SPR20 is the ECRM assigned seasonal release identifier of Spring 2020, RRS is the System Instance, and 20.1.0 represents the year (2020) and 1.0 is the RRS build number.

Example: ECRM\_SUM21\_ETMS\_21.1.1 where ECRM is the COTS S/W platform, SUM21 is the ECRM Summer 2021 release, ETMS is the System Instance being changed, and 21.1.1 is year 2021 and ETMS build number 1.1.

# 4.0 Types of Releases

Releases can be categorized into three types.

Major – new features, enhancements, capabilities, and may include accumulated bug fixes. Also known as a full release or a full build. A full release entails the delivery of all the code - not just those items that changed.

Minor – defect fixes for known problems

Emergency – unanticipated problems which make the system unavailable for use or for an urgent business need.

## 4.1 Release and Deployment Process

Please refer to the “Release Management Process Workflow” for the detailed flow chart process. The latest version can be obtained upon request from the CM team.

## 4.2 Release Management Checklist

A release management checklist is a listing of those tasks that must be accomplished for each release. The checklist is consistently applied to all releases. Below is the Release Management Checklist for SW releases:

Release Management Checklist

Instance Affected:

Release Number:

Scheduled Deployment Date and Time:

RFC Number:

Date S/W passed CAT/UAT Testing:

☐ Source code received from contractor?

☐ Delivered software posted to the X drive?

☐ Was the code given to the CM Library?

☐ Were the contractual deliverables as identified in the RFC provided to CM?

☐ Is the VDD Prepared?

☐ “Scheduled Maintenance” email coordinated and issued to the user community?

☐ Was the “Authorization to Deploy” email issued?

☐ Post Deployment testing scheduled?

☐ Post Deployment Status email issued?