|  |
| --- |
| GENUS INNOVATION LIMITED |
| Configuration Management and Release Procedure |
|  |
|  |
| **Genus** |

|  |
| --- |
| Configuration Management is to establish and maintain integrity of the product and product components throughout the Project Lifecycle. This involves identifying the configurations of configurable items throughout the product development life cycle, systematically control the changes, and maintain the configurable items integrity and traceability throughout the Product Development Life Cycle. |

Contents

[Overview 3](#_Toc447797957)

[Objective 3](#_Toc447797958)

[Scope 3](#_Toc447797959)

[Inputs 3](#_Toc447797960)

[Entry Criteria/Triggers 3](#_Toc447797961)

[Tasks 3](#_Toc447797962)

[Verification 7](#_Toc447797963)

[Guidelines 8](#_Toc447797964)

[Definitions 8](#_Toc447797965)

[Configuration Structure 9](#_Toc447797966)

[Folder Structure 9](#_Toc447797967)

[Logs 10](#_Toc447797968)

[QMS Structure 18](#_Toc447797969)

[Artifact Naming Convention 18](#_Toc447797970)

[Naming conventions for: 19](#_Toc447797971)

[Audit plan 19](#_Toc447797972)

[Back up and Restoration plan 19](#_Toc447797973)

[PEG Plan 19](#_Toc447797974)

[MPP (Schedule) 19](#_Toc447797975)

[Module Test Cases 19](#_Toc447797976)

[Metrics Baseline Report 19](#_Toc447797977)

[Training Material 19](#_Toc447797978)

[Training Strategic Plan and Tactical Plan Naming Conventions 19](#_Toc447797979)

[User Support Documents 20](#_Toc447797980)

[Document Naming Convention (If not explicitly specified) 20](#_Toc447797981)

[DAR - When using the generic template (TMPL\_DARGEN) 20](#_Toc447797982)

[PEG Audit Checklist - When using the PEG Audit Checklist template (CHKL\_ADTPEG) 20](#_Toc447797983)

[Training Audit Checklist - When using the Training Audit Checklist template (CHKL\_ADTTRN) 20](#_Toc447797984)

[IT Infrastructure Audit Checklist - When using the IT Infrastructure Audit Checklist template (CHKL\_ITAUDT) 20](#_Toc447797985)

[QMS Change Requirements (TMPL\_QMSREQ) 20](#_Toc447797986)

[Senior Management review Agenda list for PPQA - When using this template (TMPL\_PQARVW) 20](#_Toc447797987)

[Minutes of Meeting (MOM) for Support Functions (PPQA, PEG, Training, IT Infra)- when using this template (TMPL\_MINMET) 20](#_Toc447797988)

[Audit Reporting Tool for PPQA - When using this Tool (TOOL\_ADTRPT) 21](#_Toc447797989)

[Document/Artifact Revisions 21](#_Toc447797990)

[Mechanism for Document / Artifact Revision History Maintenance and Approval 21](#_Toc447797991)

[Audit Procedure 21](#_Toc447797992)

[Use of SUBVERSION 21](#_Toc447797993)

[Introduction 21](#_Toc447797994)

[SVN Server 22](#_Toc447797995)

[Client Installation 22](#_Toc447797996)

[Recommended Practices 22](#_Toc447797997)

[Responsibilities 23](#_Toc447797998)

[Configuration Administrator: 23](#_Toc447797999)

[Project Manager: 23](#_Toc447798000)

[Code Versioning 23](#_Toc447798001)

[Backup Process 23](#_Toc447798002)

[Release Process 24](#_Toc447798003)

[Technical Data Package 24](#_Toc447798004)

[Applicable Measurements 25](#_Toc447798005)

[Exit Criteria/Outputs 25](#_Toc447798006)

# Overview

Configuration Management is to establish and maintain integrity of the product and product components throughout the Project Lifecycle. This involves identifying the configurations of configurable items throughout the product development life cycle, systematically control the changes, and maintain the configurable items integrity and traceability throughout the Product Development Life Cycle.

# Objective

To identify and control the set of work products (Configurable Items) those constitute the product during the complete life cycle of the project.

# Scope

This procedure applies to all work products and development projects.

# Inputs

* Project Kickoff Email
* All Configurable Items’ creation, change and release

# Entry Criteria/Triggers

1. Project is kicked off
2. Pre-planning Schedule is prepared
3. Project Plan is initiated
4. Revision in configuration items.
5. The practitioners have undergone QMS trainings with focus on performing their processes.

\**Revisions are different instances of a particular configuration as it changes over time. Each individual and aggregate item in a configuration may be revised independently. When an item is revised all other related items also are implicitly revised.*

# Tasks

| Sr. No | Task | Owner/Role |
| --- | --- | --- |
|  | **Prepare Configuration and Data Management Plan** |  |
|  | Configuration and Data Management Plan is prepared at the start of Project Lifecycle or in the early stage of Project Planning.   * Configuration and Data Management Plan is a part of “Project Plan”. Refer TMPL\_PRJPLN | Configuration Administrator |
|  | Identify and define the users’ access rights. Refer” Master list of Configurable Items” (MSTL\_CMLIST) for Role Based Access Rights. | Configuration Administrator |
|  | Identify the tools for Storage and Retrieval of Configurable Items. Suggested tools are:   * Enterprise Project Management Tool (EPM) for storing and version controlling of the project related documents. * Visual Source Safe for storing and version controlling of the Source Code. * Subversion for storing and version controlling of the project related documents. * Subversion for storing and version controlling of the Source Code, Circuits, Drawings, Layouts.   The use of any combination of the above tools is a tailorable option that must be clearly spelled out in the plan for the project.  Ensure that the backups are provisioned for the Project Data with respect to the tools selected. Refer the Published Backup Plan for details on the same. | Configuration Administrator |
|  | Define and specify the path for the Project Repositories for Source Code, Circuits, Drawings, Layouts and project related documents. | Configuration Administrator |
|  | **Review and Approval of Configuration and Data Management Plan** |  |
|  | Get the Configuration Management Plan reviewed by Project Manager. | Configuration Administrator |
|  | Close the review findings. | Configuration Administrator |
|  | Get the approval of the Configuration Management Plan by Project Manager and PPQA Manager. Documented and approved Configuration Management Plan would be considered as the basis for Configuration Management activities. | Configuration Administrator |
|  | **Creation and Maintenance of Configuration Library** |  |
|  | Create the Configuration library structure. Ensure access rights and approval workflow. | Configuration Administrator |
|  | Refer guidelines to create the project location structure on the Configuration Management Tool. |  |
|  | Create the project location on the Configuration Management Tool for maintaining the Source Code, Circuits, Drawings, and Layouts. |  |
|  | Manage the access rights for Project Data. | Configuration Administrator |
|  | Ensure that the backups are provisioned for the Project Data with respect to the tools selected. Refer the Published Backup Plan for details on the same. | Configuration Administrator |
|  | **Configuration Control and Project Baselines** |  |
|  | Baseline and revise baselines for the project as per the base lining and revision criteria as mentioned in the Configuration Management Plan. The baselines can be revised on receipt of a change request. A valid defect report may also be considered a valid change request. | Project Manager |
|  | Amend the impacted Configurable Item. | Developer/s |
|  | Update version and revision history. | Developer/s |
|  | Update comments in the configuration management tool while checking in and/or checking out Configurable Items. | Developer/s |
|  | Ensure that proper versioning is done. | Configuration Administrator |
|  | **Release Process** |  |
|  | Prepare the Release Plan. | Project Manager |
|  | Identify and document the work products to be released. |  |
|  | Review the Release Plan and get it approved by Project Manager. | Project Manager |
|  | Upload all the work products to be released on the Server. | Project Manager |
|  | Package the work products. | Project Manager |
|  | Prepare the media for the release application. |  |
|  | Bundle the media and User Manual. |  |
|  | Prepare the “Release Note” (TMPL\_RELNOT). | Project Manager |
|  | Verify and Approve the Release Note. | Configuration Administrator |
|  | Release the Release Note. Refer Release Process Guidelines. | Project Manager |
|  | Perform Release Audit. | PPQA Members |
|  | **Technical Data Package** |  |
|  | Establish a Technical Data package. This includes the Requirement Traceability table, Design documents, Source Codes, Circuits, Drawings, Layouts etc. Refer Technical Data Package Guidelines. | Project Manager / Configuration Administrator |
|  | **Project Closure** |  |
|  | Perform Project Closure. Refer “Project Monitoring and Control” Procedure (PRCD\_PRJMAC). The principal responsibility of the Configuration Administrator at the time of closure is to conduct Configuration Audits. A configuration Audit is performed to ensure that the technical data package and the project baselines are at the defined locations and are in an ‘approved’ state. It also ensures that the configuration status of the project’s CIs are consistent. | Project Manager/ Configuration Administrator |
|  | **Project Closure Audit** |  |
|  | Identify Resources such as checklists, standards and related documents needed to perform the audit. | PPQA Members |
|  | Identify Internal Auditor from the pool of Auditors in the Organization   * Refer to the “Personnel Database”. | PPQA Members |
|  | **Perform Internal Audits and Communicate Findings** |  |
|  | Ensure the availability of the related checklists, standards and supporting documents required to conduct the audit. | PPQA Members |
|  | Conduct Audit as per the schedule. The Audit at project closure ensures that the configuration audit has been performed satisfactorily for the project, in addition to the other checkpoints. | PPQA Members |
|  | Classify findings as   * Minor / Major Noncompliance Issues (NCs) * Observations | PPQA Members |
|  | Discuss the audit findings with the Auditee and update the findings in the “Audit Non-conformance Log”. | PPQA Members |
|  | **Take Suitable Corrective Action and Close the NCs** |  |
|  | Discuss the timelines to close the NCs with the Auditee. | PPQA Members |
|  | Take suitable corrective action and close the NCs. | Auditee |
|  | Ensure that audit findings are closed. | PPQA Members |
|  | Generate “Audit Report” using “Audit Report Tool” (TOOL\_ADTRPT). | PPQA Members |
|  | Send the “Audit Report” to PEG Head for identifying Candidate Process Improvement Opportunities. | PPQA Members |
|  | Identify best practices and process improvement opportunities. Submit them to the Process Engineering Group (PEG) using “Process Improvement Proposals Database”. | PPQA Members |

\* Improvements/Suggestions are solicited on “Process Improvement Proposals Database”.  
\*For details on the Roles and Responsibilities of the practitioners, Refer "Roles and Responsibility" document in the QMS.

# Verification

* Review of Configuration and Data Management Plan by Project Manager and the PPQA Manager.
* Review of all Configurable Items for base lining by PPQA Member
* Review of Release Plan by Project Manager
* Review of Release Note by Project Manager
* Review of documents to be released
* Work Product / Release audit by PPQA Members
* Configuration Audit by PPQA Members
* Review of the process and its work products by Senior Management.

# Guidelines

Refer "Configuration Management and Release Procedure" (PRCD\_CONFIG) for Access Rights, location of work products, naming convention and types of controls.

## Definitions

**Configuration Items (CI)** – A “configuration item” is an entity designated for configuration management, which may consist of multiple related work products that form a baseline. This logical grouping provides ease of identification and controlled access.

Configuration items can be individually managed and versioned, and they are usually treated as self-contained units for the purposes of identification and change control. All configuration items are uniquely identified by names, version numbers, and other attributes.

**Library** – A library is a collection of reusable modules. There are two types of libraries viz. organization level and project level.

**Subsystem** - Subsystem refers to a collection of items that are also part of a larger system.

**Deliverables** - Deliverables or Product items are the items actually delivered to an Internal customer or a customer. They may be thought of as the top-level subsystems in a product.

**Product** - A product is the special configuration consisting of all product items. This includes executable, setup procedures, documentation, etc.

**Project** - A project is a temporary endeavor undertaken to create a unique product or service. Temporary means that every project has a definite beginning and a definite end. Unique means that the product or service is different in some distinguish way from all similar products and services.

A project consists of product deliverables plus all non-deliverable items. Source code is the principal non-deliverable, but test suites, test reports, specification documents, etc., are also candidate project items, as per the management needs of the project. The project directory is the file system sub-directory containing all project materials.

**Baseline** – A set of specifications or work products that has been formally reviewed and agreed on, which thereafter serves as the basis for further development, and which can be changed only through change control procedures.

A change to a baseline is made with a change request because it is likely to involve new agreements or contracts with other affected groups. It may also have an effect on work that has already been completed in working from the original baseline. In most cases, an accepted defect is deemed to be a valid change request.

In order to transfer a workproduct from ‘Controlled’ status to ‘Baselined’ status, indicate the same using a comment in the configuration management system. Specifically, if using SharePoint, generate a major X.0 version of the CI. For example: Functional specifications (TMPL\_FUNSPC) is baselined after approval. Whenever there is a change in specifications it will be incorporated through change management procedure and then it will be again baselined.

**Controlled Configuration Items-** Controlled Configuration Items are those, whose current version in use is known (i.e. version control), and changes are recorded maintaining a version history. For example Miniutes of meetings (TMPL\_MINMET).

## Configuration Structure

### Folder Structure

### Logs

|  |
| --- |
| \\192.168.100.9\shared\Jalaj\CMMI\screenshots\Audit Log.PNG |
| \\192.168.100.9\shared\Jalaj\CMMI\screenshots\Issue Log.PNG |
| \\192.168.100.9\shared\Jalaj\CMMI\screenshots\Personnel Database.PNG |
| \\192.168.100.9\shared\Jalaj\CMMI\screenshots\Process Improvements Proposals.PNG |
| \\192.168.100.9\shared\Jalaj\CMMI\screenshots\Project Learning.PNG |
| \\192.168.100.9\shared\Jalaj\CMMI\screenshots\ProjectCode.PNG |
| \\192.168.100.9\shared\Jalaj\CMMI\screenshots\Review Testing.PNG |
| \\192.168.100.9\shared\Jalaj\CMMI\screenshots\Schedule by Milestone.PNG |
| \\192.168.100.9\shared\Jalaj\CMMI\screenshots\Timesheet.PNG |

### QMS Structure

## Artifact Naming Convention

The Name has these Two fields, inter field separator is \_(underscore), intra field separator is -( dash)

|  |  |
| --- | --- |
| Type of doc | 6 char description |
| Policy- POLC  FORM- FORM  Template- TMPL  Chklist+ Crit- CHKL  Guideline- GDLN  Procedure- PRCD  Master- MSTL  Information- INFO  Tool- TOOL |  |

For example:

Bidirectional requirement traceability table is TMPL\_REQTRT

# Naming conventions for:

## Audit plan

Rename the template TMPL\_ADTPLN by replacing TMPL by QA-XX-XX where XX-XX represents the current fiscal year. Follow the same naming convention for “Audit Schedule” (TMPL\_ADTSCH).

## Back up and Restoration plan

Rename the template TMPL\_BKPPLN by replacing TMPL by GenusXXXX-XX where XXXX-XX represents the current Fiscal.

## PEG Plan

Rename the template TMPL\_PEGPLN by replacing TMPL by Genus yyyy-yy where yyyy-yy represents the financial year for the planned period. For eg. Genus 2010-11\_PEGPLN

## MPP (Schedule)

Name the MPP file as same as project code.

## Module Test Cases

The name has three fields. Inter field separator is \_(underscore), intra field separator is -( dash).

The first field designates the project code.

The second field is the module name.

Third field is the 6 character description of the artifact, for e.g. HWTCAS.

For eg GS1234-SMPS\_HWTCAS is the module test case document for the SMPS in the project GS1234.

## Metrics Baseline Report

Metrics Baseline Report is named as Metrics Baseline Report X where X is the sequence number of the Metrics Baseline Report.

## Training Material

Training material should be named so as to be self-explanatory.

## Training Strategic Plan and Tactical Plan Naming Conventions

Strategic plan is named as Genus yyyy-yy\_TRGSTR where yyyy-yy represents the financial year for the planned period. For eg. Genus 2010-11\_TRGSTR.

## User Support Documents

User Support Documents (eg. user manual etc.) should be named so as to be self-explanatory.

## Document Naming Convention (If not explicitly specified)

When an artifact is used in a project, its name is modified as follows:-

* The first four characters in the artifact indicating the type of document are replaced by the Project’s name/code for e.g. GS1234
* The 6 character description is left as it is( Not modified)

### DAR - When using the generic template (TMPL\_DARGEN)

* Replace the TMPL by project code, add a hyphen (-), the purpose of DAR, followed by \_DARGEN. For example: To select MPPT algorithm in project GS1234, the name will be GS1234-MPPTALGO\_DARGEN.

### PEG Audit Checklist - When using the PEG Audit Checklist template (CHKL\_ADTPEG)

* Replace the CHKL by month name, add a hyphen (-), the year, followed by \_ADTPEG. For example: For August month of 2015 year PEG audit, the name will be AUGUST-2015\_ADTPEG.

### Training Audit Checklist - When using the Training Audit Checklist template (CHKL\_ADTTRN)

* Replace the CHKL by month name, add a hyphen (-), the year, followed by \_ADTTRN. For example: For August month of 2015 year Training audit, the name will be AUGUST-2015\_ADTTRN.

### IT Infrastructure Audit Checklist - When using the IT Infrastructure Audit Checklist template (CHKL\_ITAUDT)

* Replace the CHKL by month name, add a hyphen (-), the year, followed by \_ITAUDT. For example: For August month of 2015 year IT Infrastructure audit, the name will be AUGUST-2015\_ITAUDT.

### QMS Change Requirements (TMPL\_QMSREQ)

* Replace the TMPL by PEG.

### Senior Management review Agenda list for PPQA - When using this template (TMPL\_PQARVW)

* Replace the TMPL by QA\_XXX-YY followed by \_PQARVW. For example: For August month of 2015 year, the name will be QA\_AUG-15\_PQARVW.

### Minutes of Meeting (MOM) for Support Functions (PPQA, PEG, Training, IT Infra)- when using this template (TMPL\_MINMET)

* Rename the template TMPL\_MINMET by replacing TMPL by Group name-XX-XX where XX-XX represents the current fiscal year. Group name would be QA, PEG, IT, TRAINING.

### Audit Reporting Tool for PPQA - When using this Tool (TOOL\_ADTRPT)

* Replace the TOOL by QA\_XXX-YY followed by \_ADTRPT. For example: For August month of 2015 year, the name will be QA\_AUG-15\_ADTRPT.

## Document/Artifact Revisions

Minor revision number is changed when there is a change in layout or formatting of the document, the content remaining essentially the same.

Major revision number is changed when there is a change in the content of the document.

### Mechanism for Document / Artifact Revision History Maintenance and Approval

If the document / artifact are being controlled using SharePoint then the default workflow is used for revision history maintenance and approval of the document / artifact under consideration. Revision history is maintained using check-in comments if SharePoint is being used for document control.

#### Document life cycle from its conception to approval

1. The author starts working on the document at which point it has a revision number suffixed by .1. The document is in draft state.
2. Every subsequent revision of the document increments the revision by .1 viz. 0.2, 0.3 etc. the document still remains in the draft state.
3. When the author deems the document to be final, he submits it for approval to the relevant authority. The state of the document changes to ‘Pending’ indicating that approval has been sought. The revision number remains unchanged.
4. The approver can now either approve or reject the document which changes the state to ‘Approved’ or ‘Rejected’ respectively. Approval generates a major revision of the document.

## Audit Procedure

Refer “Audit Procedure” (PRCD\_AUDITT) for details.

## Use of SUBVERSION

### Introduction

Subversion is a centralized system for sharing information. At its core is a repository, which is a central store of data. The repository stores information in the form of a file system tree - a typical hierarchy of files and directories.

Any number of clients connects to the repository, and then read or writes to these files. By writing data, a client makes the information available to others; by reading data, the client receives information from others.

The repository is a kind of file server, but it's not your usual breed. What makes the Subversion repository special is that it remembers every change ever written to it: every change to every file, and even changes to the directory tree itself, such as the addition, deletion, and rearrangement of files and directories.

When a client reads data from the repository, it normally sees only the latest version of the file system tree. But the client also has the ability to view previous states of the file system. For example, a client can ask historical questions like, “what did this directory contain last Wednesday?”, or “who was the last person to change this file, and what changes did they make?” These are the sorts of questions that are at the heart of any version control system: systems that are designed to record and track changes to data over time.

### SVN Server

The SVN server for R&D (GIL) is located at the URL: https://192.168.100.9/svn .

The configuration Administrator shall be responsible for the server administration. Administration includes, but is not limited to, repository creation, user creation, maintenance, backups and advanced support.

### Client Installation

A GUI based client, TortoiseSVN will be used for accessing and using the revision control system.

### Recommended Practices

1. The commit messages must be sufficiently detailed so as to make the history of the development self-evident.
2. Each project must be a self-contained sub repository within the root of the assigned repository.
3. Commit little and often. Commit at regular intervals at small milestones. Single commits with a huge number of changes all at once make it difficult to go back in history.
4. Only commit finished work. Never commit half-completed artifacts. This seems to go against the concept of ‘commit little and often,’ but the solution is to split the task you’re working on into manageable but logical pieces, and then commit these regularly.
5. Update regularly. This ties into the concept of ‘commit little and often.’ Perform regular updates on your working copy, to keep up to date with the changes being made in the trunk. This is important, even if it seems your current work has little to do with the rest of the team. It is also good practice to update your working copy before making any changes.
6. Do All File Operations in the Version Control System. All moves, copies and deletes must be done using the version control to avoid conflicts during the next update.
7. Do not put executables, project files, workspace files etc. under version control. As a rule of thumb, Version control must only be used for files that are edited manually.
8. For non-text based file formats such as MS-Office files, Images etc., it is recommended to get an exclusive lock before editing, as merging binary files is almost impossible.
9. Learn when to use, or not to use, each of the 18 basic version control operations using Tortoise SVN. The operations are Create (Create a new, empty repository.); Checkout(Create a working copy).; Commit (Apply the modifications in the working copy to the repository as a new changeset.); Update (Update the working copy with respect to the repository.); Add (Add a file or directory.); Edit (Modify a file.); Delete (Delete a file or directory.); Rename (Rename a file or directory.); Move (Move a file or directory.); Status (List the modifications that have been made to the working copy.); Diff (Show the details of the modifications that have been made to the working copy.); Revert (Undo modifications that have been made to the working copy.); Log (Show the history of changes to the repository.); Tag (Associate a meaningful name with a specific version in the repository.); Branch (Create another line of development.); Merge (Apply changes from one branch to another.); Resolve (Handle conflicts resulting from a merge.); Lock (Prevent other people from modifying a file.)
10. Share other best practices with others and the Administrator, so that others can benefit from your experience.

## Responsibilities

### Configuration Administrator:

* Administrate the Configuration Management tool
* Ensure that developers are trained in the use of the Configuration Management tool
* Establish and maintain the Configuration Management Plan and the Configuration Management Procedures documentation.
* Comply with the responsibilities listed in Configuration Management Procedures.

### Project Manager:

* Review the Configuration Management Plan and Configuration Management Procedures documentation
* Comply with Responsibilities listed in Configuration Management Procedures

## Code Versioning

For informal baselines the Configuration Administrator must be notified of the change/s.

Any changes in Source codes must be tested and built by the developer. Integration and verification baselines must be approved by the Project Manager.

## Backup Process

Refer “Backup Procedure” (PRCD\_BACKUP) for details on Backup, Archival and Restoration.

## Release Process

All releases from Design to Integration and Validation contain the following information:

* What is in the release
* Which stage / Phase, the release is being provided to and when
* The media the release is on
* Any known problems in the release
* Any known fixes in the release
* Installation instructions

There are several product release types. The following is an example of some the product release types:

*Internal Release* - This release is only for releasing to internal test organizations such as Quality Assurance. This release should not be distributed to the outside vendors, since it may have errors and the functionality of the product is not complete.

*Beta Release* - This release is for testing by internal test organizations as well as for the outside vendors. The purpose of this release is to allow customers to test the product in their environment and help in finding problems which cannot be detected in-house since it is not possible to create every customer’s environment.

*Release* -The official release of the product to customer.

## Technical Data Package

Contents of Technical Data Package include:

|  |  |
| --- | --- |
| Sr. No. | TDP Contents |
|  | Requirement Documents (Customer requirements, Functional specifications, Requirement Traceability Table) |
|  | Planning Data (Project Plan, Risk Plan, Estimates, Schedule) |
|  | Source Codes, Schematics, BOMs, Mechanical drawings, PCB layouts |
|  | Design Documents |
|  | Test Cases, Test Plans |
|  | Audit Reports |
|  | Metrics Reports |
|  | Defect List |
|  | Risk Matrix |
|  | Support Documents |
|  | Setup Files(Executables) |

# Applicable Measurements

NA

# Exit Criteria/Outputs

* Configuration and Data Management Plan is reviewed, approved and distributed to relevant stakeholders.
* Approved Release Plan
* Approved Release Note
* Configuration Audit Report
* Work Product/Release Audit Report
* Released Package