|  |
| --- |
|  |

**rossypro:private:var:folders:_h:dfhb65ms5xj_17c_z9xt5tvh0000gn:T:TemporaryItems:msoclip:0:clip_image001.gif  
ATLAS TDAQ < sous-groupe >**

rossypro:private:var:folders:_h:dfhb65ms5xj_17c_z9xt5tvh0000gn:T:TemporaryItems:msoclip:0:clip_image002.gif

**< élément > Exigences en matière**

Document Version : 1.0

ID de document : ATLAS-TDAQ-2002-XXX

Du document Date : Date de dernière modification

Document statut : Projet | Final

rossypro:private:var:folders:_h:dfhb65ms5xj_17c_z9xt5tvh0000gn:T:TemporaryItems:msoclip:0:clip_image003.gif

Résumé

Résumé texte pour le document sur les exigences.

Instituts et auteurs :

Institut 1 : A. auteur, b. auteur, auteur de C.

Institut 2 : Auteur A.

Institut 3 : A. auteur, auteur de B.

etc....

**Tableau 1**Document changement Record

|  |  |  |  |
| --- | --- | --- | --- |
| **Titre :** | ATLAS TDAQ < sous-groupe > < Élément > exigences | | |
| **ID :** | ATLAS-TDAQ-2002-XXX | | |
| **Version** | **Question** | **Date** | **Commentaire** |
| 1 | 1 |  |  |

*Un numéro de version différent conviendrait au document si des modifications de fond au contenu du document ont été apportées. Numéros de question différente au sein d’une version donnée indiquent des changements mineurs seulement comme les corrections orthographiques et grammaticales.*

*Note sur le texte en rouge : le texte rouge fournit des explications sur le contenu suggéré pour chaque section ainsi que quelques exemples.*

# 1 Introduction

*Bref texte d’introduction sur le composant.*

**1.1 But du document**

Ce document expose les exigences de la < élément > de l’ATLAS TDAQ/DCS < sous-groupe > . Ils seront la base pour la conception et la mise en œuvre de la < élément > dans le contexte du système ATLAS Trigger DAQ/contrôleurs de domaine.

*Inclure une description très courte de ce que le composant est responsable (lignes 3-5).*

**1.2 Glossaire, acronymes et abréviations**

*Fournir des définitions pour tous les termes, acronymes et abréviations utilisées dans ce document. Des glossaires existants qui sont utilisés doivent être référencées, pour éviter une répétition ici.*

**1.2.1 Glossaire**

*Pour obtenir une description longue, utiliser une paire de formats de paragraphe de Dfn1Part1Term/Dfn1Part2Desc.*

***Exemple : ferme***

*Un ensemble de nœuds informatiques reliés par un réseau ou un bus*

**Quelque chose d’autre à définir**

Et voici la description de quelque chose d’autre

**1.2.2 Sigles et acronymes**

*Pour les courtes descriptions utiliser une paire de formats de paragraphe de Dfn1Part1TermR1H/Dfn1Part2Desc :*

**CERN**Laboratoire européen de physique des particules

**ASA**Un autre acronyme idiot

**1.3 Références**

*La liste de tous les documents externes référencés dans ce document*

1

2

3

4

5

# 2 Description générale [facultatif]

*Cette section est facultative. Pour les composants simples, tous la description nécessaire pourrait figurer dans l’introduction. Cependant, il devrait inclure de brèves déclarations portant sur les informations décrites dans les sous-sections suivantes.*

**2.1 Cadre de la < élément >**

*Décrire les sous-systèmes et les systèmes externes. Description générale du contexte dans lequel le composant sera exploitée. Description de systèmes externes avec lesquels la composante devrait permettre d’interagir. Description des interfaces de systèmes externes avec le composant.*

**2.2 Capacités générales de < élément >**

*Décrire les principales capacités requises et pourquoi ils sont nécessaires.*

**2.3 Contraintes générales sur < élément >**

*Décrire les principales contraintes qui s’appliquent et pourquoi ils existent.*

**2.4 Dépendances et les hypothèses générales**

*Décrire les hypothèses que le composant doit faire concernant les systèmes externes. Par exemple, les services le composant nécessite des systèmes externes pour mettre en œuvre.*

**2.5 Caractéristiques de l’utilisateur**

*Décrire qui utilisera le composant et quand.*

# 3 Des contraintes spécifiques, hypothèses/dépendances, cas d’utilisation et exigences

*Pour une petite composante, un seul niveau de description peut être suffisant dans chaque catégorie. Pour un composant complex, une liste hiérarchique en option au sein de chaque catégorie peut être utile. Cela commencerait avec par exemple, les contraintes concernant le composant entier, suivi d’une dépression, dans les sous-sections des contraintes pour les divers volets auxiliaires.*

**3.1 Contraintes**

*Une contrainte est quelque chose qui touche la façon que les exigences sont satisfaites. Il impose des restrictions sur la conception du système qui n’affectent pas le comportement extérieur du système, mais qui doivent être remplies pour satisfaire aux techniques ou obligations du projet. Contraintes typiques sont argent, temps, interaction avec déjà existant, technologie des systèmes.*

*Chaque contrainte doit posséder un identifiant unique (ne pas utiliser les numéros de position ou de numérotation automatique dans Word car les nombres doivent rester le même après des ajouts ou des suppressions) afin de faciliter le suivi à travers les phases ultérieures du processus de développement logiciel. Un titre pour la contrainte peut être ajouté, ou laissé en blanc. La rigueur avec laquelle la contrainte doit être appliquée doit être indiquée par l’utilisation correcte des termes « doit », « devrait », « mai », « peut » dans l’instruction constraint (voir liste de vérification des exigences OnlineSW pour plus d’informations : http:// de**satddoc.cern.ch/Altas/DaqSoft/ / i**nspect/Requirements\_checklist.html). Contraintes essentielles sont non négociables ; d’autres peuvent être moins vitalement important et soumis à négociation. Toute autre information pertinente (p. ex., les exigences connexes) peut être ajoutée dans une note.*

*CO001*Un exemple :*Restriction du temps*

*Le < élément > sera opérationnelle le 1st mars 2002*

*CO002*

*Un exemple : Le logiciel < élément > doit être écrits en C++.*

**CO [id] titre de contrainte [facultatif]**

[Instruction constraint]

Notez les informations supplémentaires facultatives...

**3.1.1 < sous-volet 1 > contraintes [facultatifs]**

*CO101*Un exemple :*Restriction du temps*

*The <sub-component 1> shall be operational on 1st January 2002.*

**3.1.2 <sub-component 2> Constraints [optional]**

**CO201**

**3.2 Assumptions and Dependencies**

*An assumption/dependency is a requirement or constraint the component puts on an external system with which the component will interact. The external system has to implement these in order for the component to fulfil its role correctly. N.B. It is important that these Assumptions and Dependencies are made known to those responsible for the design and development of the relevant external components.*

*Each assumption/dependency must have a unique identifier (do NOT use heading numbers or autonumbering in Word as the numbers must remain the same after any additions or deletions) to facilitate tracing through subsequent phases of the software development process. A title for the assumption/dependency may be added, or left blank. The strictness with which the assumption/dependency is to be applied shall be indicated by the correct use of the terms "shall", "should", "may", "can" in the assumption/dependency statement (see OnlineSW Requirements Checklist for further information: http://atddoc.cern.ch/Altas/DaqSoft/sde/inspect/Requirements\_checklist.html). Essential assumptions/dependencies are non-negotiable; others may be less vitally important and subject to negotiation. Any other relevant information (e.g. related requirements) can be added in a note.*

*AD001*

*External component shall deliver the run number to <component> at the start of each new run.*

Note Optional extra information…

**AD [id] Assumption/Dependency Title [optional]**

[Assumption/Dependency Statement]

Note Optional extra information…

**3.2.1 <sub-component 1> Assumptions and Dependencies [optional]**

**AD101 Assumption/Dependency [optional]**

[Assumption/Dependency Statement]

Note Optional extra information…

**3.2.2 <sub-component 2> Assumptions and Dependencies [optional]**

**AD201**

**3.3 Use Cases [optional]**

*Use cases capture the different things an external "user" wishes to do with the component. The user may be a human or external software/hardware. A use case describes a sequence of actions, performed by the component that yields a result of a value to the user. The use case describes how users and the system work together to achieve a particular goal. Use cases can be a useful starting point for deriving functional requirements and as illustrations and examples of the components behaviour. The inclusion of the use case section is optional but authors are encouraged to put in use cases if they have them already prepared.*

*Each use case must have a unique identifier (do NOT use heading numbers or autonumbering in Word as the numbers must remain the same after any additions or deletions) to facilitate tracing though subsequent phases of the software design process. A title for the use case may be added, or left blank. A short description of the use case should be given. Any other relevant information (e.g. related requirements) can be added in a note.*

**UC001 Use Case Title [optional]**

[Use Case Description]

Note Optional extra information…

***UC [id]*An Example: *Starting the system***

*An operator will use the graphical user interface to request that <component> starts all child processes.*

Note Optional extra information…

**3.3.1 <sub-component 1> Use Cases [optional]**

**UC101 Use Case Title [optional]**

[Use Case Description]

Note Optional extra information…

**3.3.2 <sub-component 2> Use Cases [optional]**

**UC201**

**3.4 Functional Requirements**

*A functional requirement is something that the component must do. It captures the intended behaviour of the component. The behaviour may be expressed as services, tasks or functions the <component> is required to perform.*

*Each requirement must have a unique identifer*

*Each requirement must have a unique identifier (do NOT use heading numbers or autonumbering in Word as the numbers must remain the same after any additions or deletions) to facilitate tracing through subsequent phases of the software design process.A title for the requirement may be added, or left blank. A statement of the requirement must be given. Essential requirements shall be marked as such by the correct use of the terms "shall", "should", "may", "can" (see OnlineSW Requirements Checklist for further information: http://atddoc.cern.ch/Altas/DaqSoft/sde/inspect/Requirements\_checklist.html). Essential requirements are non-negotiable; others may be less vitally important and subject to negotiation. Each requirement shall include a measure of priority so that the developer can decide the production schedule. Any other relevant information, such as examples to illustrate the purpose of the requirement can be added in a note.*

**UR001 Requirement Title [optional]**

[UR Statement]

Priority [Priority for incremental delivery]

Note Something…

*UR002*An Example: *Starting child processes*

*The <component> software shall provide a means to start all child processes.*

Priority High

**3.4.1 <sub-component 1> Functional Requirements [optional]**

**UR101 exigence de titre [facultatif]**

[Instruction UR]

Priorité [priorité de livraison supplémentaire]

A noter quelque chose...

**3.4.2 Exigences fonctionnelles < sous-volet 2 > [facultatif]**

**UR201**

**3.5 Exigences non fonctionnelles**

*Une exigence non-fonctionnelle est une propriété que du composant doit avoir. Ils sont attributs système supplémentaires. Il existe plusieurs catégories d’exigences non fonctionnelles : performance, interface, opérationnel, ressources, vérification, test d’acceptation, documentation, sécurité, les exigences en matière de portabilité, de qualité, de fiabilité, de maintenabilité et de sécurité (voir Guides de génie logiciel, C.Mazza et al., Prentice Hall, p.99 pour plus d’informations). Qu’ils spécifient comment bien certains aspects comportementaux du système devrait être mise en oeuvre. Si vous le souhaitez, le style « Exigences de la rubrique » permet de regrouper les exigences dans les différentes catégories.*

**UR0XX exigence de titre [facultatif]**

[Instruction UR]

Priorité [priorité de livraison supplémentaire]

A noter quelque chose...

**Exigences de performance [facultatifs]**

*UR [id]*Un exemple :*Performances*

*Tous les processus enfants seront mis dans un délai de 1 minute.*

Priorité élevée

Exigence connexe remarque UR002

**Exigence d’interface s [facultatif]**

**Dépenses opérationnelles [facultatifs]**

**Resource Requirements [optional]**

**etc.**

**3.5.1 <sub-component 1> Non-Functional Requirements [optional]**

**UR1XX Requirement Title [optional]**

[UR Statement]

Priority [Priority for incremental delivery]

Note Something…

**3.5.2 <sub-component 2> Non-Functional Requirements [optional]**

**UR2XX**

This document has been prepared using the Requirements Document Template provided and approved by the Atlas TDAQ and DCS Connect Forum. For more information, go to

<http://atlas-connect-forum.web.cern.ch/Atlas-connect-forum/>.

This template is based on the SDLT Single File Template that have been prepared by the IPT Group (Information, Process and Technology), IT Division, CERN (The European Laboratory for Particle Physics) and then converted to MS Word. For more information, go to http://framemaker.cern.ch/.