|  |
| --- |
| PTC |
| Multi Context Based Agreements |
| Design Note |

|  |
| --- |
| pghiya@ptc.com  10/24/2017 |

|  |  |  |  |
| --- | --- | --- | --- |
| Design Note | | | |
| File Location | Place the Design Note under Source Code Management control   * in PDS: [Windchill Security Team > Folders > Designs](https://pds.ptc.com/Windchill/app/#ptc1/tcomp/infoPage?ContainerOid=OR%3Awt.projmgmt.admin.Project2%3A629374363&oid=OR%3Awt.folder.SubFolder%3A1039787121&u8=1)> Multi-Context-Design-Note | | |
| Release | Eagle | Owning Team | Security WPD |
| Status | In Work | Product Owner | Laura Nissen |

# References and Supporting Documents

|  |  |
| --- | --- |
| **Document** | **Title/Link** |
| Capability Packages | [Release Theme: 626951](http://integrity.ptc.com:7001/im/issues?selection=626951) |
| Backlog Item(s) | [Story: 7055370](http://integrity.ptc.com:7001/im/issues?selection=7055370) |
| Related Design Notes, Approaches, or Specifications | [Approach - Multi-Context Context Level Agreement](http://integrity.ptc.com:7001/im/issues?selection=6808044) |
| SPR(s) | None |
| Dependencies | None |

# Reviewers

|  |  |  |
| --- | --- | --- |
| **Review Team** | | |
| Architecture Rep | Sarang Joshi | <Review Instructions> |
| Scrum Master | Sarang Joshi | <Review Instructions> |
| Technical Lead | Pritam Ghiya / Chintan Anand | <Review Instructions> |
| Technical Lead |  | <Review Instructions> |
| Architecture Rep |  | <Review Instructions> |
| Product Owner | Laura Nissen | <Review Instructions> |
| CAS team |  |  |
| QA Representative | Snehal Dange / Ajinkya Mhaskar | <Review Instructions> |
| PUBS Representative | Manank Gor | <Review Instructions> |

# Design Issues

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Dates**  **(mm/dd/yyyy)** | **Name** | **Description** | **Resolution** | **Status**  **Open/**  **Closed** |

# Introduction

## Background

Today the Context Based Agreements can be created on any context, For example on a product context or project or organization. The Agreement will be applied to all the objects inside the container, where it is created.

It is challenging, for the customers who are having 1000s of contexts (Either projects or products or libraries etc.) where similar kind of agreement is to be created at multiple contexts. In this case creation, maintaining of numerous similar agreements is also very hard.

**Problem Statement**

The goal of this Release theme is to provide a provision to create a Multi Context Based Agreement objects for the user, where a single agreement instance can be applied to multiple contexts.

**Components**

This new feature should take care of all below listed functionalities:

* Agreement UI Operations – Create, Edit, View
* Related Agreements View
* Access Permission check API (has Access)

## Scope and Assumptions

The scope of this design note focuses on the approach provided by 6808044– Multi-Context Context Level Agreement Approach Note. This solution describes facilitating the feature of Multi-Context Context Level Agreements.

Below Assumptions are considered:

1. For the Context Based Agreement, its mandatory to select at least one context from the context picker.
2. If customer needs to add additional restriction on a particular context, then they will create new agreement for that context. For example, consider “MultiContextAgreement1” is created for 10 contexts including “Demo Product”. It has date range as 1 Sep 2017 to 3 Jan 2018. Now customer needs to restrict “Demo Product” to date range 1 Jan to 3 Jan. Customer will create new agreement for this context and his use case will be satisfied.
3. The Contexts listing in the context picker, while creating the Context based agreement with multiple authorized contexts, will be the current context and child contexts of that container on which user has “Read” permission.

# Design

## Overview

The design is majorly divided into 3 sections here:

1. UI Operations:

This section includes, design for UI Changes for all below actions:

* Create Agreement
* Edit Agreement
* View Agreement
* Related Agreements View

1. Server Side:

This section defines the changes to incorporate the consideration of multi-context level agreements for evaluation of authorization as a part of hasAccess API. This will also include the respective change in tier-3 caching to accommodate the contexts references as authorized object.

## Internal Design

This section describes in more detail the internal design for all the 3 sections.

1. **UI Operations:**

* **Create Agreement View:**

In the Create Agreement Window of Windchill at any context level, on selection of Context Level Agreements the Authorized Objects Wizard tab is removed from the window. For this RT, We need to include an Authorized Contexts wizard tab. This Window should have context picker. The selected contexts should be shown in Table of this tab. On submitting the form, the data is save to DB in AgreementAuthorizedObjects Table.

List of Changes:

* Security/src\_web/netmarkets/jsp/agreements/create.jsp – inclusion/exclusion of wizard step for authorizedContexts
* Security/src\_web/config/actions/Agreements-actions.xml - Add an action for the new wizard step
* Add a jsp for the above action at path Security/src\_web/netmarkets/jsp/agreements/
* Add a Component builder java class for above jsp inorder to build a table for this tab
* Add Action model for the above table in ‘Security/src\_web/config/actions/Agreements-actionModels.xml’ file. For this use case only 2 actions are applicable (Add (+) , Delete(-))
* The actions defined in action model should be defind in Agreements-Actions.xml
* The corresponding jsp page should be created in Security/src\_web/netmarkets/jsp/agreements/. For example, the jsp page for add action should have a context picker.
* The table names, action icons, names, description etc., data should be defined in agreementResource.java
* agreements.js – function for resetting the AuthContextStep
* CreateAgreementFormProcessor.java - In Post Process method, collect the authorized contexts and persist in DB.
* AgreementsCommands – Have helper method in this class to implement addAuthorizedContexts.

addAuthorizedObjects method need to handle both (Authorized Objects and Context)

* **Edit Agreement View**

For the edit agreement, the new tab wizard should be added, and this is step should be inserted conditionally for the context based agreement. The related contexts can be fetched in the table builder, which is configured as a component builder in the jsp of this wizard step.

Listing the files to be changed:

* Security/src\_web/netmarkets/jsp/agreements/edit.jsp – inclusion of wizard step for editing authorizedContexts if the agreement is a context based agreement.
* Security/src\_web/config/actions/Agreements-actions.xml - Add an action for the new wizard step
* Add a jsp for the above action at path Security/src\_web/netmarkets/jsp/agreements/
* Add a Component builder java class for above jsp in order to build a table for this tab
* Same action model, that is used on the jsp of new wizard tab for create agreement view will be applicable for this jsp page as well.
* The table names, action icons ,names , description etc., data should be defined in agreementResource.java
* EditAgreementFormProcessor.java - In Post Process method, persist the authorized contexts, similarly how authorized objects are handled.
* **View Agreement**

The info page for the agreements is provided by AgreementsInfoBuilder.java class. This is the component builder class for the info page of authorized agreement class. ‘FullContainerAgreementsInfoPageTabset’ tab set and ‘third\_level\_nav\_full\_container\_agreement’ Navigation bar is set for context based agreements. In order to include to show authorized contexts in this view below changes are required:

* For ‘FullContainerAgreementsInfoPageTabset’ definition in the Agreements-actionModels.xml file, add a sub model for authorizedcontexts.
* Add Definition for the above specified sub model in the same xml.
* Add action in the above definition, in order to list authorized context with in the definition of model.
* Define the action in Agreement-actions.xml.
* Create a table builder java class as component builder for the defined action above. Fetch the authorized contexts for the agreements in this builder.
* third\_level\_nav\_full\_container\_agreement’ model is set as navigation bar. ‘agreeFullContainerGeneral’ is the sub model included and this definition holds the actions. Add action defined for listing the authorized contexts in the agreeFullContainerGeneral model definition.
* Include the table names in the resource files for the newly defined actions, tables.
* **Related Agreements:**

The Table for the related agreements table, get the list of agreements related from ‘***AgreementDataModelManager.getAuthorizationAgreements(securityLabeledObj).*** This API should be changed in order to fetch the Multi Context Based agreements. This method first collects the container reference of provided object. With the help of this container id, and agreement type the list of authorized agreements are fetched from DB. Changes should be made in AgreementDataModelManager class.

1. **Server Side:**

While persisting retrieving and viewing authorized context, exiting authorizedObjects (For Standard Agreement) API, existing Data utilities, and Server Utility methods will be utilized. It will be modified to handle both (Objects and Contexts). This could be achieved by converting Context in AgreementAuthorizedObject and using the same.

While executing hasAccess API, agreements are evaluated. Within this API the agreement meta data is cached and evaluated. Change should be made here in order to include the multi context agreements while evaluating. The method ‘isAuthorizationGrantedByAgreement’ in StandardAgreementService.java is the place where the agreement is evaluated for the object.

AgreementAuthorizedObjects table now holds only business objects. It will now hold the contexts. System should be able to deal with both single context agreements (maintained in older format) and multi context agreement. Once any single context agreement is modified, it will add its single context in AgreementAuthorizedObjects and start behaving as per new design.

For Related Agreement table, query have been modified to fetch contexts associated with multi-context agreements.

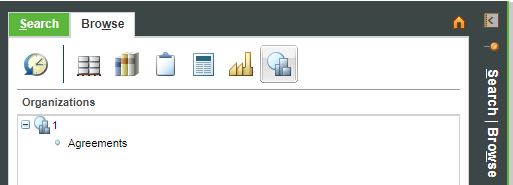
Feature to enable non-Org user to access Org -Agreement Cabinet

As per the [RT](http://integrity.ptc.com:7001/im/issues?selection=626951) Multi-Context Context Level Agreement, User should able to create single agreement applicable to multiple contexts. While creating multi-context agreement user can only select the current context or child context. With this restriction a Agreement Manager who have access to only selected product may not add other context as they may not be the child of current context. The solution to it is, create Agreement at Organization’s Agreement cabinet so that it could add all the available contexts under this Organization. As we can’t expose Organization to non-Org user, we have introduced a work around mechanism.

This mechanism is property controlled. Property name is “wt.access.agreements. nonOrgUserAcessToSuperAgreementCabinet”. In OOTB, this will be set to “false” by default. If user want to use this mechanism, they need to set it to “true”.

As per this mechanism, When an Organization Agreement Cabinet will be created, an adhoc permission (READ, MODIFY, CREATE, and MOVE) will added for Agreement Manager group. This will enable non-Org user (Must be part of Agreement Manager group) to access the Organization Agreement Cabinet without exposing other details. This will also enable them to manage folders(Create, Move) within Agreement Cabinet.

In order to access this Org Agreement Cabinet, Org Admin need to provide the direct URL of the Org Agreement Cabinet to the Agreement manager. This will be one time activity, once the URL is being hit by Agreement Manager, adhoc permissions will be set and this Agreement Cabinet will be visible to all the Agreement Managers of that Org. There will be no further need to use the URL for accessing Org Agreement Cabinet, it will be visible as below to Agreement Managers, they can navigate and access it.



In case, Organization Agreement Cabinet already created, once non org user will hit this URL, same adhoc permission will be created.

#### API changes

* AgreementMetaDataCacheEntry will have to hold ArrayList of containers.
* StandardAgreementService.isAuthorizationGrantedByAgreement will have to check the object context with all contexts of the agreement.
* AgreementDataModelManager.getContextBasedAuthorizationAgreements will have to return applicable both types of agreements. This result will be shown in Related Agreements table.

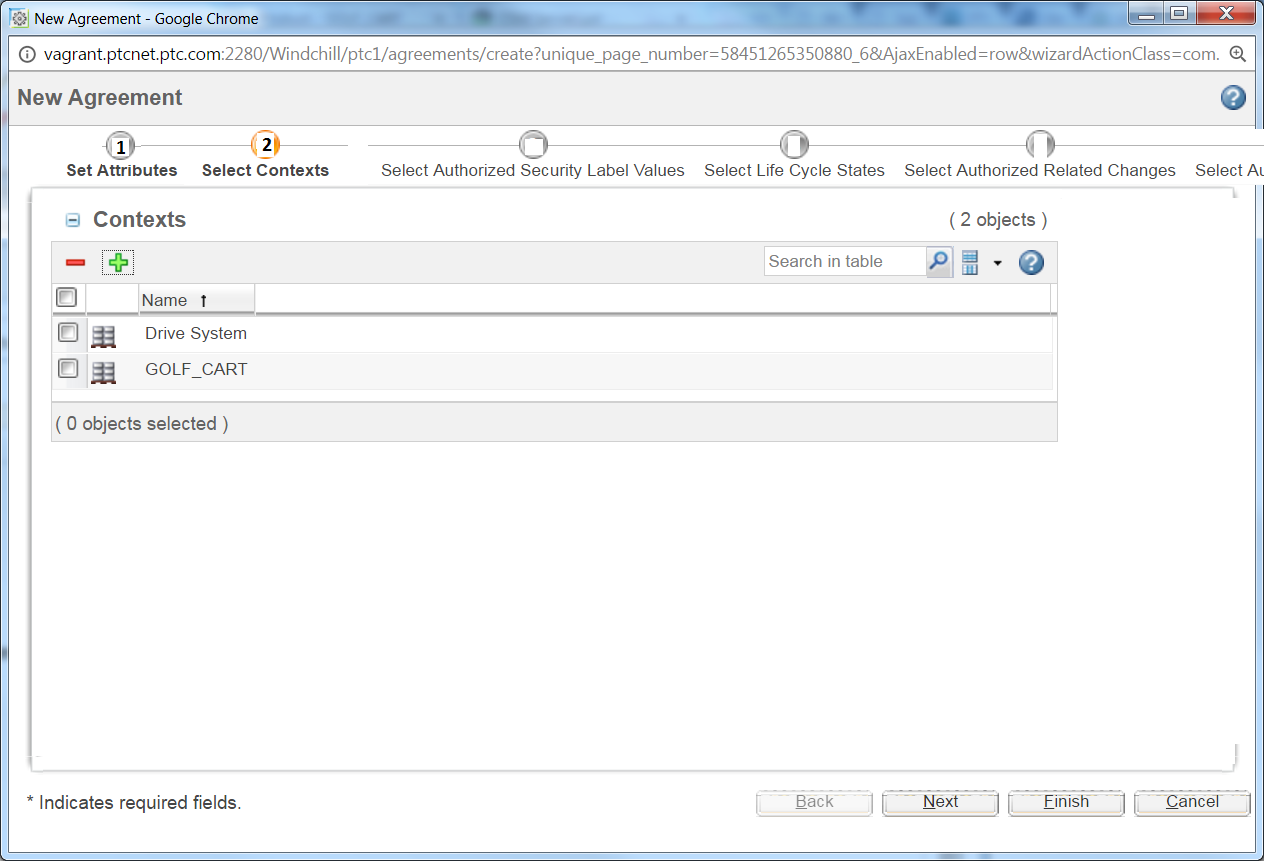
## User Experience

### UI Implementation Considerations

#### New Agreement UI

Similar functionality can be found in creation of Replication Package.

If ContextBasedAgreement is selected, new tab will be shown at second step. This will allow to select multiple contexts. This window should display all low level contexts for the user to multi select.



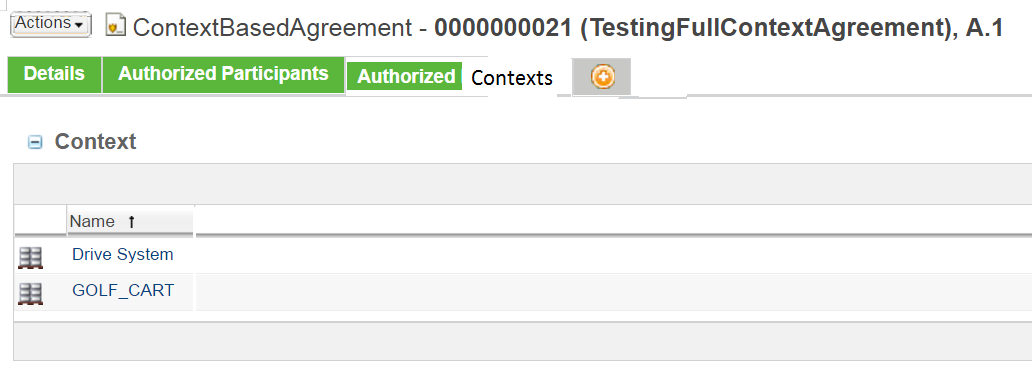
It is mandatory to select at least one context.

Similar tab will be seen for edit agreements. If single context agreement is edited it will be converted to multi-context agreement.

**Sample code from replication package:**

* [selectContextStep](http://bla-opengrok/search?q=selectContextStep&project=wnc-main&defs=&refs=&path=-src_test+-src_selenium+&hist=)
* [replicationPackage.contextCriteria.table](http://bla-opengrok/search?q=%22replicationPackage.contextCriteria.table%22&project=wnc-main&defs=&refs=&path=-src_test+-src_selenium+&hist)

#### View Agreements \ Info page



It will have a new tab “Authorized Contexts”

Existing single context agreements will show its current context in this table.

[***AgreementsInfoBuilder***](http://bla-opengrok/s?refs=AgreementsInfoBuilder&project=wnc-main) will have to be updated

**Sample code from replication rule:**

* [repl.criteriaInfo.context](http://bla-opengrok/search?q=%22repl.criteriaInfo.context%22&project=wnc-main&defs=&refs=&path=-src_test&hist=)
* [replicationRuleTabSet](http://bla-opengrok/search?q=replicationRuleTabSet&project=wnc-main&defs=&refs=&path=-src_test&hist)

#### Security

No specific security issue has been anticipated.

# Performance Impacts

Performance testing has to be done with 4000 contexts, but could be fine-tuned after discussion with PO based on the actual customer’s scenario.

# Upgrade and Update

No upgrade impact as both agreements will can co-exist in system.

# Alternate Considerations

None