

c.a.p.e. IT KIX extension
"KIXMaintenancePlan"
Installation and configuration

c.a.p.e. IT[®] GmbH

Version: 1.0.0

Imprint

Title:	c.a.p.e. IT KIX extension "KIXMaintenancePlan"
Topic:	Installation and configuration
Author:	c.a.p.e. IT® GmbH
Keywords:	Maintenance, KIX
Comments:	
Filename:	KIXMaintenancePlan_installation_and_configuration_en .odt
Version:	1.0.0
Set and Layout:	LibreOffice.org Version 4.2.8.2

Table of contents

1	General advices.....	5
1.1	Document information.....	5
1.1.1	Objectives.....	5
1.2	Changes.....	5
1.2.1	Change history.....	5
1.2.2	Authors.....	5
1.2.3	Document owner.....	5
1.2.4	Authorized staff.....	5
1.3	List of abbreviations.....	5
2	Installation.....	6
2.1	Requirements.....	6
2.2	Package Installation.....	6
2.3	New Access Control Lists (ACL).....	6
2.4	New ticket type.....	6
2.5	New dynamic fields.....	6
3	Configuration.....	8
3.1	SysConfig Options.....	8
3.1.1	Classes for time based maintenances.....	8
3.1.2	Classes for threshold based maintenances.....	8
3.1.3	Relevant deployment states.....	8
3.1.4	Check threshold based maintenances.....	9
3.1.5	Link type.....	9
3.1.6	Replace current version.....	9
3.1.7	Default ticket data.....	9
3.1.8	Ticket attribute – CI attribute mapping.....	9
3.1.9	Default article data.....	10
3.1.10	Scheduler registration.....	10
3.1.11	Validity check event.....	10
3.1.12	Validity check parameters for maintenance plan.....	10
3.1.13	Validity check parameters for maintained CIs.....	11
3.1.14	Event to check on value update for threshold based.....	11
3.1.15	Write ticket data back to CI.....	11
3.1.16	CI attribute – Dynamic field mapping.....	11
3.1.17	Similar tickets search parameters.....	12
3.1.18	Permission for change popup.....	12
3.1.19	Set required lock.....	12
3.1.20	Change several ticket attributes and dynamic fields.....	12
3.1.21	Dynamic fields for maintenance tab.....	12
3.1.22	Dynamic field groups for maintenance tab.....	13
3.1.23	Dynamic fields – group mapping for maintenance tab.....	13
3.1.24	Event to add an external note automatically.....	13
3.1.25	Subject for the external note.....	13
3.1.26	Body for the external note.....	13

3.2 Usage of KIXMaintenancePlan.....	14
3.2.1 First steps after installation.....	14
3.2.2 Configure a maintenance.....	14

1 General advices

1.1 Document information

1.1.1 Objectives

This document contains all the information to install and configure the c.a.p.e. IT extension "KIXMaintenancePlan".

1.2 Changes

1.2.1 Change history

Version	Date	Changed chapters	Short description	Changed by
1.0.0	04.03.16	All	document creation	Ricky Kaiser
1.0.0	10.03.16	All	Typos and optimization	Rene Boehm
1.0.0	12.05.16	3.1.8	Title, key and description	Ricky Kaiser
1.0.0	17.05.16	3.1.15, 3.2.2	Title, key, description, usage	Ricky Kaiser
1.0.0	07.06.16	2.5, 3.1.24, 3.1.25, 3.1.25	4 new fields; added chapters	Ricky Kaiser
1.0.1	01.11.16	2.5	Changed some mistakes	Ricky Kaiser

1.2.2 Authors

Surname, first name	Organisation	Function
Boehm, Rene	c.a.p.e. IT® GmbH	Head of Product Development
Kaiser, Ricky	c.a.p.e. IT® GmbH	Developer

1.2.3 Document owner

Organisation	Surname, first name	Address	Contacts
c.a.p.e. IT® GmbH		Schönherrstr. 8 09113 Chemnitz	info@cape-it.de +49 371 27095 - 620

1.2.4 Authorized staff

Surname, first name	Organisation	Function	Auth.
Boehm, Rene	c.a.p.e. IT® GmbH	Head of Product Development	rw
Kaiser, Ricky	c.a.p.e. IT® GmbH	Developer	rw

1.3 List of abbreviations

CLI	Command Line Interface
GUI	Graphical User Interface
CMDB	Configuration Management Database
CI	Configuration Item

2 Installation

2.1 Requirements

To install and use the KIX extension "KIXMaintenancePlan" your installation environment needs to fulfill the following requirements (or higher versions):

- a KIX 16.0.0 installation

2.2 Package Installation

Just install the package "KIXMaintenancePlan" via the Package Management in the KIX GUI or use the CLI package manager. Please make sure all the requirement (see 2.1) are fulfilled.

2.3 New Access Control Lists (ACL)

During installation, three new ACLs are added. They hide some ticket function for maintenance tickets and the maintenance ticket function and tab for tickets which are not maintenance tickets. The third ACL hides the new type in the customer frontend.

2.4 New ticket type

During installation, a new ticket type ("Maintenance") is created. It is used to mark the maintenance tickets. Also the access control list will only work with this type by default.

2.5 New dynamic fields

During installation, several new dynamic fields are created to store the relevant maintenance information:

Field name	Function
MaintenancePlan	Contains the ID the relevant maintenance plan CI.
MaintenanceDate	Contains the date when the maintenance is done.
MaintenanceCI	Contains the ID of the relevant maintained CI.
MaintenanceCIClass	Contains the class of the relevant maintained CI.
MaintenancePlanTime	Contains the planned time for this maintenance.
MaintenanceCILocation	Contains the location ID of the relevant maintained CI.
MaintenanceCostCenter	Contains the corresponding cost center ID of the relevant maintained CI.
MaintenanceCIContact	Contains the responsible customer user ID of the relevant maintained CI.
MaintenanceCIType	Contains the type of the relevant maintained CI.
MaintenanceSupplier	Contains the customer/supplier which have to do the maintenance.
MaintenanceDueDate	Contains the due date of a time based maintenance.
MaintenanceThreshold	Contains the threshold of a threshold based maintenance.
MaintenanceThreshold Attribute	Contains the relevant attribute name with the current operating counter value of the maintained CI.
MaintenanceReachedValue	Contains the current operating counter value of the relevant maintained CI.

MaintenanceType	Contains the type of the relevant maintenance of the maintained CI.
MaintenanceCosts	Contains the costs of the maintenance.
RepairCosts	Contains the repair costs of the maintenance.
MaintenancePlannedStart	Contains the planned maintenance start date/time.
MaintenancePlannedEnd	Contains the planned maintenance end date/time.
MaintenanceActualStart	Contains the actual maintenance start date/time.
MaintenanceActualEnd	Contains the actual maintenance end date/time.

See chapter 3.1.8, to know or specify which dynamic field gets its value from which CI attribute.

3 Configuration

3.1 SysConfig Options

After you have installed the package, you can adjust the basic parameters if necessary.

To configure the extension just open the SysConfig in the admin area. Select the SysConfig group "KIXMaintenancePlan". After the page reload select one of the subgroups:

- Core::CMDBMaintenanceCheckScheduler
- Core::ITSMConfigItem
- Core::Ticket
- Frontend::Agent::KSBSimilarTicketList
- Frontend::Agent::Ticket::ViewMaintenanceDocumentation
- Frontend::Agent::Ticket::ViewTicketZoomTabMaintenance

A default configuration is already included in the package. You can change it to your needs. The following sections list the available SysConfig options.

3.1.1 Classes for time based maintenances

SysConfig Option: CMDBMaintenanceCheckScheduler::CIClassesTimeBased

Defines the relevant CI classes (Key) with the maintenance attributes. Content can be 1 ("NextCheckDateReminder" is used to check whether a maintenance ticket have to be created) or 0 ("NextCheckDate" is used).

Please note:

This options is required. The default value "MaintenanceCI" is just a placeholder. You have to provide the relevant class(es) you want to use (see chapter 3.2.1).

3.1.2 Classes for threshold based maintenances

SysConfig Option: CMDBMaintenanceCheckScheduler::CIClassesThresholdBased

Defines the relevant CI classes (Key) with the maintenance attributes. Content can be 1 ("NextCheckThresholdReminder" is used to check whether a maintenance ticket have to be created) or 0 ("NextCheckThreshold" is used).

Please note:

This options is required. The default value "MaintenanceCI" is just a placeholder. You have to provide the relevant class(es) you want to use (see chapter 3.2.1 for more information).

3.1.3 Relevant deployment states

SysConfig Option: CMDBMaintenanceCheckScheduler::CIClassesDeploymentStates

Defines which deployment states (separated by comma or semicolon) are relevant for which CI class to be checked whether a maintenance ticket have to be created. Only CIs with these deployment states are considered for the checks.

Please note:

This options is required. The default value "MaintenanceCI" is just a placeholder. You have to provide the relevant class(es) you want to use (see chapter 3.2.1 for more information).

3.1.4 Check threshold based maintenances

SysConfig Option: CMDBMaintenanceCheckScheduler::CheckThresholdBased

Specifies whether threshold based maintenances should be checked by the scheduler task. This can cause more than one maintenance ticket for the same maintenance if the check runs again and the parameters to be checked are still matching.

Please note:

The default value is "Yes".

3.1.5 Link type

SysConfig Option: CMDBMaintenanceCheckScheduler::TicketLinkType

Defines the link type between the maintenance ticket and its corresponding CI.

Please note:

The default value is "RelevantTo" and this option is required.

3.1.6 Replace current version

SysConfig Option: CMDBMaintenanceCheckScheduler::ReplaceCIVersion

Specifies whether the change of the new "NextCheckDate" or "NextCheckThreshold" value replaces the current CI version (Yes) or creates a new one (No).

Please note:

The default value is "Yes".

3.1.7 Default ticket data

SysConfig Option: CMDBMaintenanceCheckScheduler::NewTicket###TicketData

Ticket data of the newly created ticket. It is also used as fallback. Key can be a CI class followed by "::::" and the ticket attribute.

Please note:

This option is required.

3.1.8 Ticket attribute - CI attribute mapping

SysConfig Option:

CMDBMaintenanceCheckScheduler::NewTicket###TicketDataCIAttributeMapping

Defines a mapping which ticket attribute (Key) get its value from which CI attribute (Content). Key can be a CI class followed by "::::" and a ticket attribute or a dynamic field (use "DynamicField_FieldName"). The Content have to be "Maintenance::::" followed by the attribute if the attribute is a sub-attribute of the maintenance attribute in the relevant maintained CI classes or it is the maintenance attribute itself. The class is used if the specific assignment should only be valid for the given class.

Please note:

This option is required. It is recommended to use CI reference attributes (e.g. 'ServiceReference') if the relevant ticket attribute is not a dynamic field. Only dynamic fields of type "ticket" are possible.

3.1.9 Default article data

SysConfig Option: CMDBMaintenanceCheckScheduler::NewTicket###ArticleData

Article data of the newly created ticket. It is also used as fallback. Key can be a CI class followed by "::" and the article attribute.

Please note:

This option is required.

3.1.10 Scheduler registration

SysConfig Option:

Daemon::SchedulerCronTaskManager::Task###CMDBMaintenanceCheckScheduler

This is options is the scheduler registration. When it runs it checks all maintenances of the relevant CIs and creates maintenance tickets if necessary. The content for key "Schedule" is a cron notation.

Please note:

This option is not required but you should not deactivate this option unless you have to. It is also necessary that the daemon runs. The default value for the key "Schedule" means "every day at 3 o'clock (am)".

3.1.11 Validity check event

SysConfig Option: ITSMConfigItem::EventModulePre###100-CheckAttributes

This is the registration for a ConfigItem event module which checks the content of ConfigItem attributes BEFORE a new version of the ConfigItem is created. Use the following options "CheckCIAttributes###CIAttributeValuesMaintenancePlan" and "CheckCIAttributes###CIAttributeValuesMaintenanceCI" to specify the parameters for the check. This event also ensures that every maintenance plan is only used once in a maintained CI.

3.1.12 Validity check parameters for maintenance plan

SysConfig Option: CheckCIAttributes###CIAttributeValuesMaintenancePlan

Defines special CI attributes of the class MaintenancePlan for the validity check. Key is the attribute followed by "::" and a specific value of it. The content are other attributes which have to be mandatory if the specific value is used. If the check gets a negative result, an error message will be replied.

Please note:

This option is required. By default it specifies that the attribute "Interval" has to be filled if "Time based" is used for the maintenance type and that "Contact" must have a content if the customer user type is set to "fixed".

3.1.13 Validity check parameters for maintained CIs

SysConfig Option: CheckCIAttributes###CIAttributeValuesMaintenanceCI

Defines special CI attributes of the given CI class for the validity check. Key is the class followed by "###", a maintenance sub-attribute followed by ":::" and a specific value of it. The content are other attributes which have to be mandatory if the specified value is used. Use "Maintenance:::" before the mandatory attribute to mark it as sub-attribute of the currently handled maintenance of the relevant CI. If a class is not specified, the specific entry is used for every given class in the options "CMDBMaintenanceCheckScheduler::CIClasses..." (see chapters 3.1.1, 3.1.2 and 3.1.3). If the CI does not have a value for the specified sub-attribute of the currently considered maintenance, the corresponding attribute of the relevant maintenance plan CI is checked. If this also fails, an error message will occur.

Please note:

This option is required. By default it specifies that the attribute "Interval" and "NextCheckDate" of the currently considered maintenance have to be filled if "Time based" is used for the maintenance type and that "ThresholdAttribute" and "NextCheckThreshold" of the currently considered maintenance must have a content if the maintenance type is set to "Threshold based".

3.1.14 Event to check on value update for threshold based

SysConfig Option: ITSMConfigItem::EventModulePost###100-ThresholdCheck

This is the registration for a ConfigItem event module. It checks whether a maintenance ticket has to be created for threshold based maintenances of the edited CI. Some of the relevant options in "Core::CMDBMaintenanceCheckScheduler" will be used (see chapters 3.1.1, 3.1.2, 3.1.3, 3.1.5, 3.1.7, 3.1.8 and 3.1.9).

3.1.15 Write ticket data back to CI

SysConfig Option: Ticket::EventModulePost###600-WriteMaintenanceTicketDataIntoCI

If a dynamic field is updated in a maintenance ticket, the value will be given to the corresponding CI. Key 'ReplaceVersion' determines whether the current version should be replaced or a new version have to be added for the attribute update. The next option "UpdateMaintenanceCIData###CIAttributeMapping" is used to provide the relevant dynamic fields and CI attributes.

3.1.16 CI attribute – Dynamic field mapping

SysConfig Option: UpdateMaintenanceCIData###CIAttributeMapping

Defines a mapping which CI attribute get its value from which dynamic field. Key has to be the dynamic field and the content can be the attribute or "Maintenance:::" and the attribute if the attribute is a sub-attribute of the maintenance attribute or the maintenance attribute itself.

Please note:

This option is required. Only dynamic fields of type "ticket" are possible.

3.1.17 Similar tickets search parameters

SysConfig Option: KIXSideBarSimilarTickets###000 (010, 020)

Defines some attributes to determine which tickets are similar. These tickets must have the same value for the specified attribute. The content can be 1 (should be same) or 0 (is ignored). For every option (000, 010, 020) a tickets search is performed and the results are combined into one list.

Please note:

The attributes (dynamic fields) "MaintenancePlan" and "MaintenanceCIClass" have to be the same, regardless of the specified attributes in these options. Do not change the keys. Other keys are not possible here.

3.1.18 Permission for change popup

SysConfig Option: Ticket::Frontend::AgentTicketMaintenance###Permission

With this option you can define the required ticket permissions to change the configured attributes (see option 3.1.20) within the maintenance documentation popup.

Please note:

This option is required and the default value is "rw".

3.1.19 Set required lock

SysConfig Option:

Ticket::Frontend::AgentTicketMaintenance###RequiredLock

With this option you can configure whether a ticket lock is required to use the maintenance documentation popup. If the ticket isn't locked yet, the ticket gets locked and the current agent will be automatically set as its owner. This has impact on the search of similar tickets if "SameOwner" is set to 1 (new owner = owner for search, see chapter 3.1.17).

Please note:

The default value is "Yes".

3.1.20 Change several ticket attributes and dynamic fields

SysConfig Option: Ticket::Frontend::AgentTicketMaintenance###...

With these options you can configure whether a user can change the relevant ticket attribute in the maintenance documentation popup.

Please note:

The dynamic field "MaintenanceDate" should be active in this documentation popup, because of the event listener in chapter 3.1.15. So if a user documents the result of the maintenance with this screen, he/she can also set the date and the event listener will write it back to the corresponding CI.

3.1.21 Dynamic fields for maintenance tab

SysConfig Option:

Ticket::Frontend::AgentTicketZoomTabMaintenance###DynamicField

This option specifies the dynamic fields shown in the maintenance ticket tab. The Key is the dynamic field and the content can be 0 = disabled or 1 = enabled.

3.1.22 Dynamic field groups for maintenance tab

SysConfig Option:

Ticket::Frontend::AgentTicketZoomTabMaintenance###DynamicFieldGroups

This option defines the dynamic field groups for the maintenance ticket tab. The key is the priority of the group for ordering. The content is the name of the group.

Please note:

This option is required.

3.1.23 Dynamic fields – group mapping for maintenance tab

SysConfig Option:

Ticket::Frontend::AgentTicketZoomTabMaintenance###DynamicFieldsGroupMapping

This option assigns the dynamic fields to the relevant group in the maintenance ticket tab. The Key is the dynamic field and the content should be a key (priority) from option "Ticket::Frontend::AgentTicketZoomTabMaintenance###DynamicFieldGroups".

Please note:

This option is required.

3.1.24 Event to add an external note automatically

SysConfig Option:

Ticket::EventModulePost###700-CreateExternalNoteOnPlannedStartUpdate

This option add automatically an external note to a maintenance ticket if the dynamic field "MaintenancePlannedStart" is updated in this ticket. The Key "RelevantTicketType" is used to ensure that the ticket with the updated field is a maintenance ticket.

3.1.25 Subject for the external note

SysConfig Option: CreateExternalNoteOnPlannedStartUpdate###Subject

Defines the subject of the external note. Placeholders for ticket attributes are possible (e.g. "<TICKET_TicketNumber>" or "<TICKET_DynamicField_NameOfField>").

3.1.26 Body for the external note

SysConfig Option: CreateExternalNoteOnPlannedStartUpdate###Subject

Defines the body of the external note. Placeholders for ticket attributes are possible (e.g. "<TICKET_TicketNumber>" or "<TICKET_DynamicField_NameOfField>").

3.2 Usage of KIXMaintenancePlan

3.2.1 First steps after installation

At first choose your CI classes which should use the functions of this extension. These classes have to be extended with some additional attributes. In the file "Maintenance.def" in the folder "var/packagesetup/InitialCIClassDefinitions/" you can find the attributes that you have to add to the class definitions..

There are two new class reference attributes. You have to remove or change them to your existing configuration. "CILocation" is an attribute to link the relevant CI with a location (building, room, ...), to know where this CI is physically located. If a existing CI class in your configuration is used as "location" but has another name you have to change the value "ReferencedCIClassName" of the "CILocation" attribute to the name of your "location" class. Similarly, you can proceed with the attribute "AssignedCostCenter". If want to use this attribute but you have no "cost center" class, you can use the default definition in the folder mentioned above to create a new class.

Next you have to adjust three dynamic fields. Open the edit screen of the dynamic fields "MaintenanceCostCenter", "MaintenanceCILocation" and "MaintenanceCI". There you have to select the relevant CI classes which should be recognized by the field and save your changes.

Finally, you have to adjust three SysConfig option (see chapters 3.1.1, 3.1.2 and 3.1.3). There you must fill in the relevant classes you have chosen in the first step.

3.2.2 Configure a maintenance

The functionality of this extension consists of three main parts. The ticket which contains the maintenance information and should be used for maintenance documentation. For each configured maintenance a ticket will be created if the conditions to be tested are true (e.g. today is the date for the maintenance – time based maintenance). The second part is the maintenance plan CI. It is used as a blueprint for the maintenances in the relevant CIs. The last but not least part is the CI to be maintained itself.

First you have to create the maintenance plan CI. There you can specify which type of maintenance will be used (time based or threshold based). For each type there are some additional attributes you should fill with useful content. These attributes are grouped, so open the relevant group with the plus button. You can also define a ticket template, which should be used for ticket creation. With the attribute "Customer user type" you can specify which customer user should be the ticket customer user:

Type	Value
ConfigItem responsible	The user specified in the attribute "Responsible user" in the maintenance CI.
Cost Center responsible	The user specified in the attribute "Responsible" in the linked cost center of the maintenance CI.
fixed	The user given in the attribute "Contact".

The attribute "Maintenance supplier" is a reference to the customer company object. With the attribute "Ticket attachment" you can add attachments which will be added to the ticket, too.

Please Note:

Each attachment attribute can only handle one attachment, so be careful. If you use the upload button of a existent attachment attribute again, you will replace the attachment than add one.

Class: MaintenancePlan

* Name:
 * Deployment State:
 * Incident State:
 * Maintenance type:
 Time based:
 Threshold based:
 Ticket-Template:
 * Customer user type:
 Contact (if fixed):
 * Maintenance supplier:
 Ticket attachment: Keine Datei ausgewählt.
 Ticket attachment:
 Planned maintenance time (min):
 Note:
 Attachment: Keine Datei ausgewählt.
 Submit

After this create a new CI of one of the classes you have chosen for the maintenance feature. In the new attribute "Maintenance" you can select the relevant maintenance CI plan with the help of autocomplete. Like in the maintenance plan itself you have to fill some additional attributes in the relevant attribute group. You have also the possibility to overwrite the preferences of the maintenance plan (e.g. to select another maintenance type). You will receive an error message if some required attribute values are missing. You can also define more then one maintenance for a CI (up to 10), but a specific maintenance plan can only be used once.

If everything is configured the event listener (see chapter 3.1.14) and the daemon (3.1.10) will do the checks and create maintenance tickets if necessary.

In a maintenance ticket all relevant data about the maintenance is bundled in the new tab "Maintenance Information".

Use the new ticket function "Maintenance Documentation" to document the result of the maintenance. Depending on which dynamic fields are enabled for this function and which fields you defined in option "UpdateMaintenanceCIData###CIAttributeMapping" (see chapter 3.1.16) the content of these fields will be given to the corresponding CI.

With the sidebar "Apply changes also to..." you can apply your changes and notes within this function to other similar tickets (see chapter 3.1.17). By default the article, ticket state, pending time, priority, owner, responsible, lock state and the content of dynamic fields will be given to the tickets.