SAP SOLUTIONS FOR GOVERNANCE, RISK, AND COMPLIANCE

How-To Guide

Document Management System in GRC Process Control 2.5

SAP GRC Regional Implementation Group

Applicable Releases:

SAP Process Control 2.5

IT Practice / Topic Area:

SAP Process Control 2.5 ,DMS

IT Scenario / Capability:

SAP Process Control 2.5 ,DMS

October 2008

Document History

Document Version	Description
1.12	Updated with content server allias confifuration and Tom Burger's comment
1.11	Recommend to add the sizing and Transport path (may be in next release)
1.10	Draft version

Typographic Conventions

Type Style	Description
Example Text	Words or characters quoted from the screen. These include field names, screen titles, pushbuttons labels, menu names, menu paths, and menu options. Cross-references to other documentation
Example text	Emphasized words or phrases in body text, graphic titles, and table titles
`Example text`	File and directory names and their paths, messages, names of variables and parameters, source text, and names of installation, upgrade and database tools.
`Example text`	User entry texts. These are words or characters that you enter in the system exactly as they appear in the documentation.
`<Example text>`	Variable user entry. Angle brackets indicate that you replace these words and characters with appropriate entries to make entries in the system.
`EXAMPLE TEXT`	Keys on the keyboard, for example, F2 or ENTER.

Icons

Icon	Description
	Caution
	Note or Important
	Example
	Recommendation or Tip

Table of Contents

1. Introduction

The Document Management System within the Process control 2.5 application is used for maintaining document administration models in the Knowledge Provider infrastructure. The Knowledge Provider (KPro) is a component of the SAP Web Application Server and is the central service for administrating and storing documents and document-like objects. For example attachment, URL link etc.

Attachment is a ***document*** or ***URL link*** stored within Process control documents storage and attached to some Process control object (according to its key).

This guide describes how the attachment or URL links are stored within the Process control 2.5 application, various attributes of the attachment, document monitoring, important DDIC objects,ABAP Programs, Function modules and also illustrates the hints for debugging the attachments, important transaction codes related to document management system and document transport.

2. Prerequisites

- GRC Process Control 2.5 needs to be installed with NWBC (Netweaver Business Client) configuration.

- /sap/bc/contentserver node is activated as external alias section of transaction SICF.

3. Document storage model in PC2.5

3.1 Storage of Attachments

The document storage for the PC2.5 application works following way:

Attaching document to a Process control object or any other object stored in case management or in plan tables

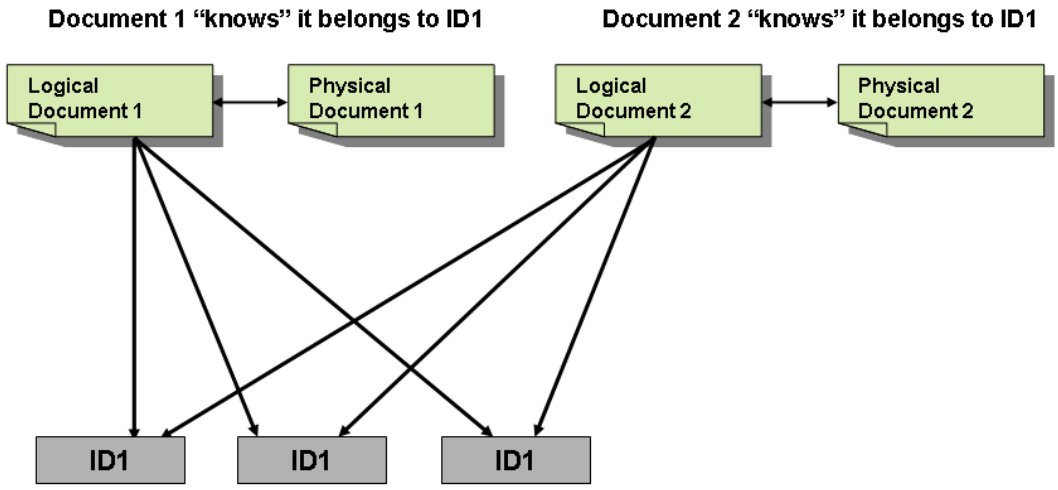
Objects

- stored in HR tables
- stored in case management or in plan tables
- time-dependent, can exist for several validity periods
- identified by object type and object ID
- have no connection to "their" attachments (documents and links)

Attachments

- stored in KPro
- time-independent
- know to which Process control Object they belong to

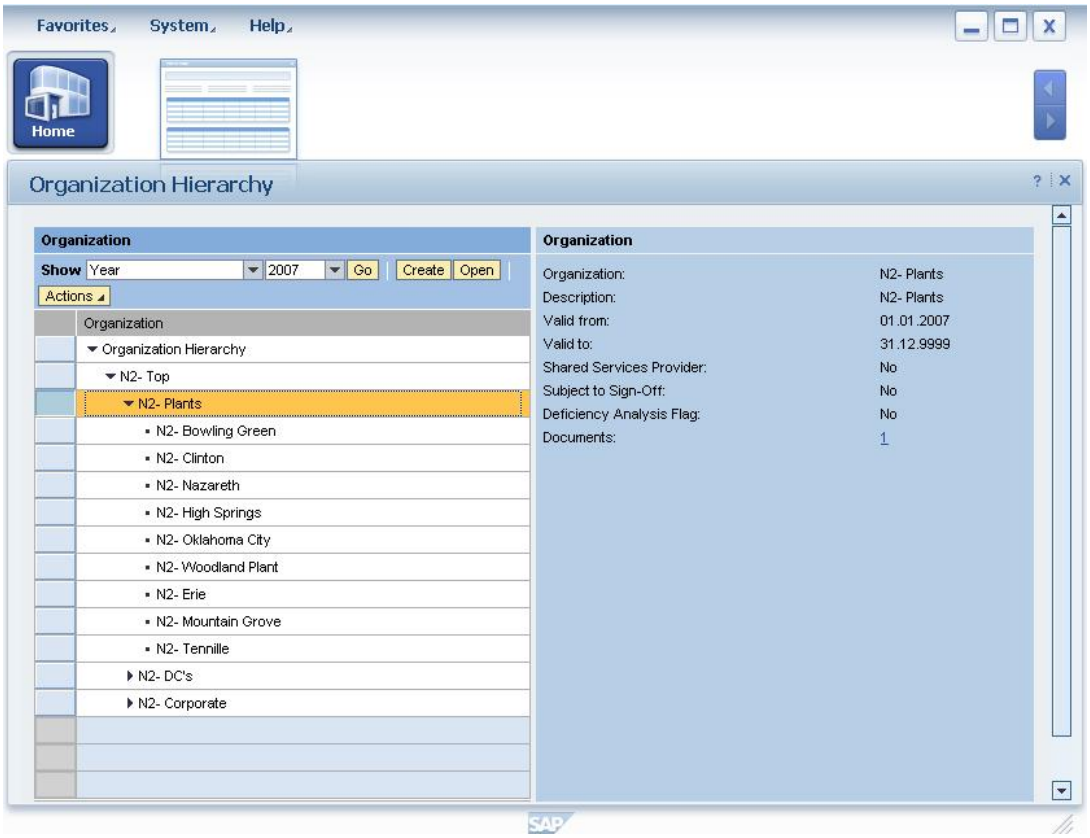
In the below picture the Process Control Object ID1 doesn't know what are the documents are attached to it. But Document 1 and Document 2 have the information which Process control object they are attached to, through document management model.



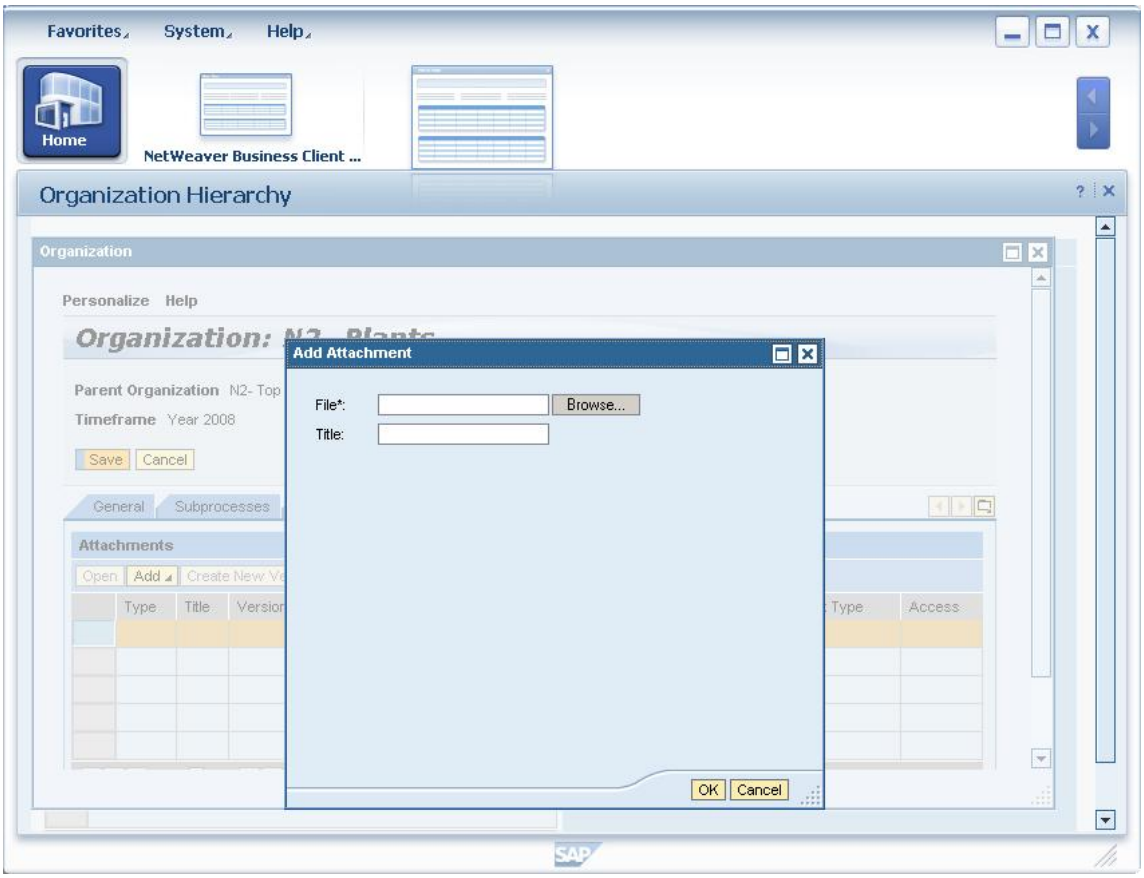
Process control Objects ID1 doesn't know about the attachments.

Adding a document in PC2.5 for a specific organization through NWBC

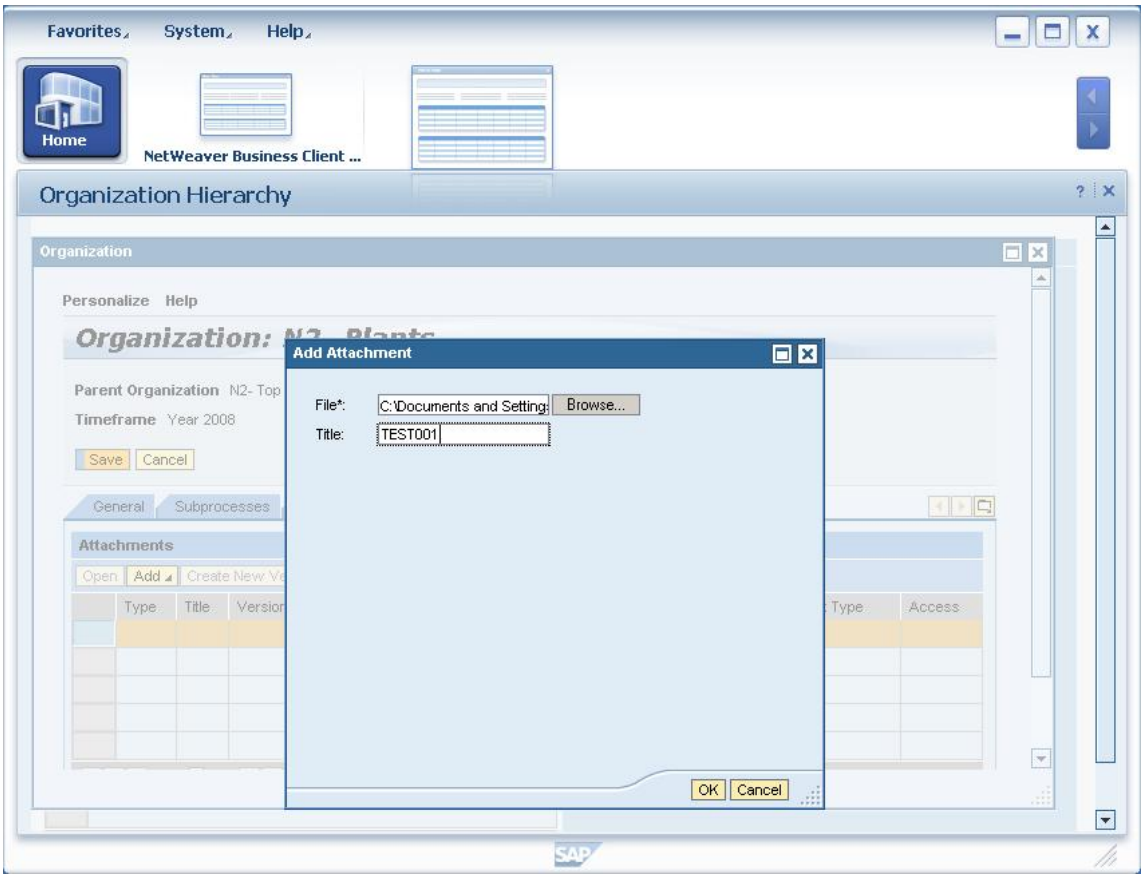
- Click on Organization Link and select the organization



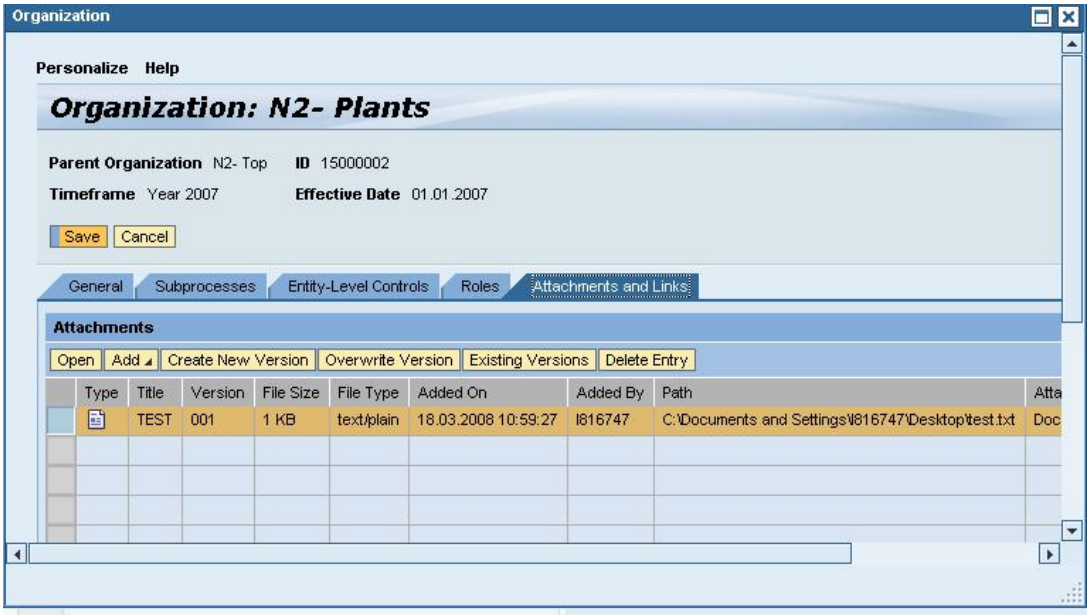
- Click on the open button then go to the Attachment and Links tab ;To add an attachment click on the add button and select the file to add to the organization



- Select the file and enter a title. Then hit ok button.



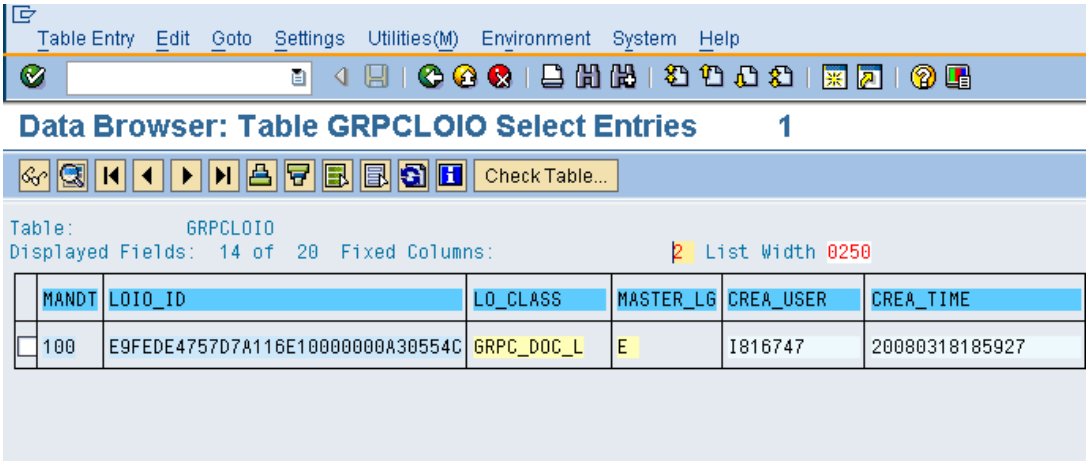
- The file will appear in the attachment window of the organization.



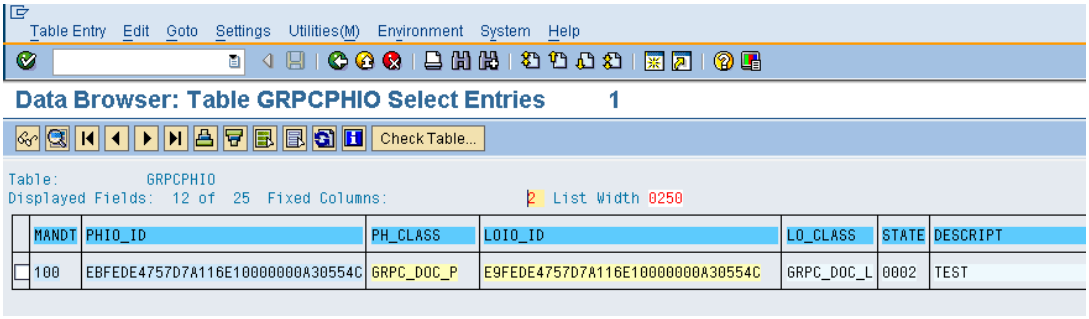
Effect on Back end

- The document ids (Logical and physical) and document class will be stored in the following tables:

GRPCLOIO Instances of Logical Information Objects



GRPCPHIO Instances of Physical Information Objects



3.2 GRPC Document Model

In the transaction **dmwb** there is defined '**GRPC**' documents model. There are relevant document classes (logical and physical) with their corresponding standard and specific attributes/properties. From this GRPC documents model following DB tables were generated (they hold records about each attachment in MIC application):

- **GRPCCHKF** - Outgoing Hyperlinks from Physical Objects
- **GRPCCHKO** - Files for Physical Information Objects
- **GRPCLOIO** - Instances of Logical Information Objects
- **GRPCLOIOT** - Incoming Relationships of Logical Information Objects
- **GRPCLOPR** - Descriptions of Logical Information Objects
- **GRPCLORE** - Attribute Values of Logical Information Objects
- **GRPCLORI** - Outgoing Relationships of Logical Information Objects
- **GRPCPHF** - Incoming Relationships of Physical Information Objects
- **GRPCPHHR** - Use of Target Anchors in Physical Objects
- **GRPCPHIO** - Instances of Physical Information Objects
- **GRPCPHNM** - Outgoing Relationships of Physical Information Objects
- **GRPCPHPR** - Checkout Data for a Physical Information Object
- **GRPCPHRE** - Attributes of Physical Information Objects
- **GRPCPHRI** - File Name of Last Checkout

Those tables listed above are generated by the TA ***dmwb*** from corresponding documents model (in this case GRPC). These are used automatically by KPro runtime once some operation is requested on documents of document classes defined in this GRPC document model.

In advance each physical document has an attribute named '**STORAGE_CATEGORY**' which is pre-filled for document classes with value 'GRPC_DB'. This attribute holds so called ***storage category***. Storage categories are maintained by transaction **OACT**. In this transaction there is mapping defined between storage category and ***documents storage***. Documents storage is then defined by transaction **OAC0**. The default (to customer delivered) storage is SAP DB storage called also 'GRPC_DB' (same name as storage category). The DB storage consists of following DB tables:

GRPCCONT1 -GRPC: Table for Document Contents (Import/Export)

Remark: If document property '**STORAGE_CATEGORY_MAINT**' is set then it's possible to re-define in transaction **SKPR08** this mapping to different storage category (per document class), so newly created documents of given document class have this new storage category.

Remark 2: Customer is allowed to create it's own documents storage (in TA OAC0) i.e. on dedicated 3rd party SAP supported content server (i.e. Documentum) and the by using mapping transaction between storage categories and storages (TA OACT) redirect the PC2.5 attachments to this content server.

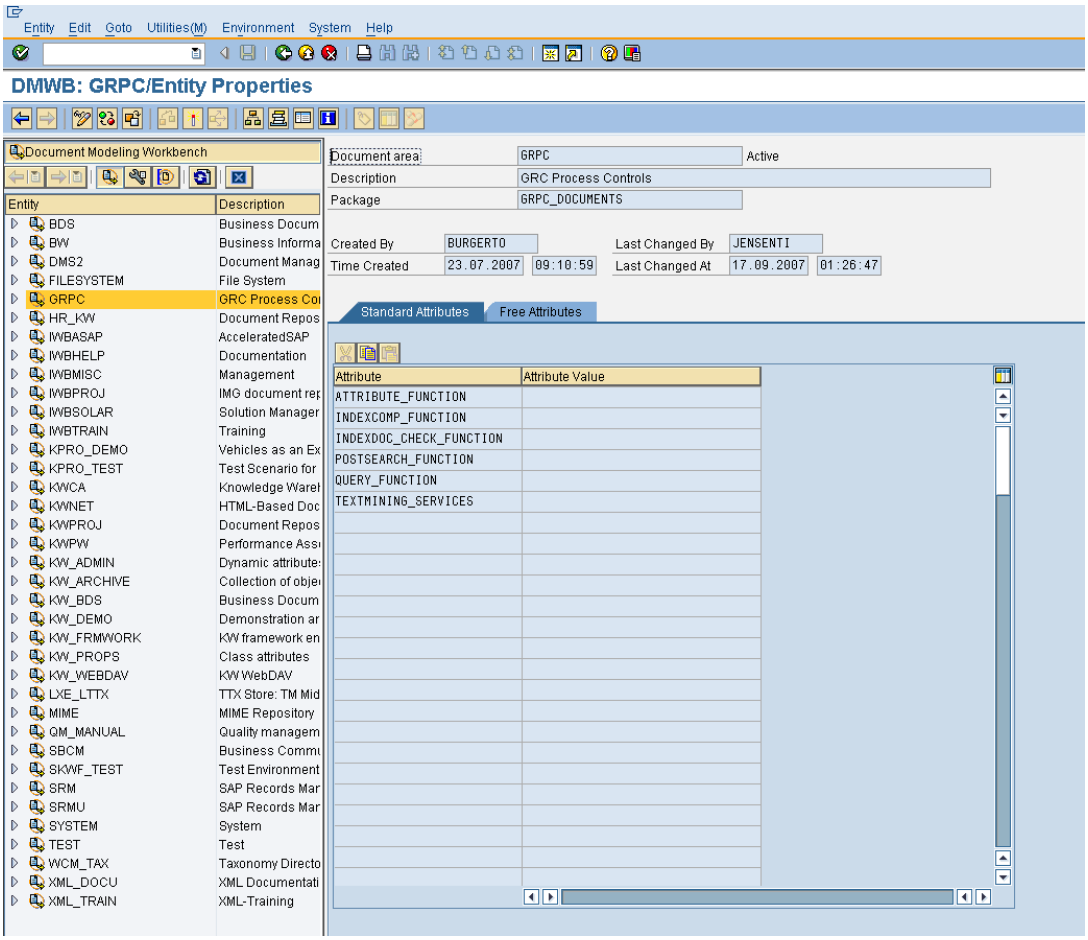
In the TA ***dmwb*** following document classes are defined

- **GRPC_DOC_L - Logical class – GRC PC documents**
- **GRPC_URL_L - Logical class – GRC PC URL Links**
- **GRPC_DOC_P - Physical class – GRC PC documents**
- **GRPC_REA_P - Physical class - Reporting Attachments**
- **GRPC_URL_P - Physical class - GRC PC Hyperlinks**

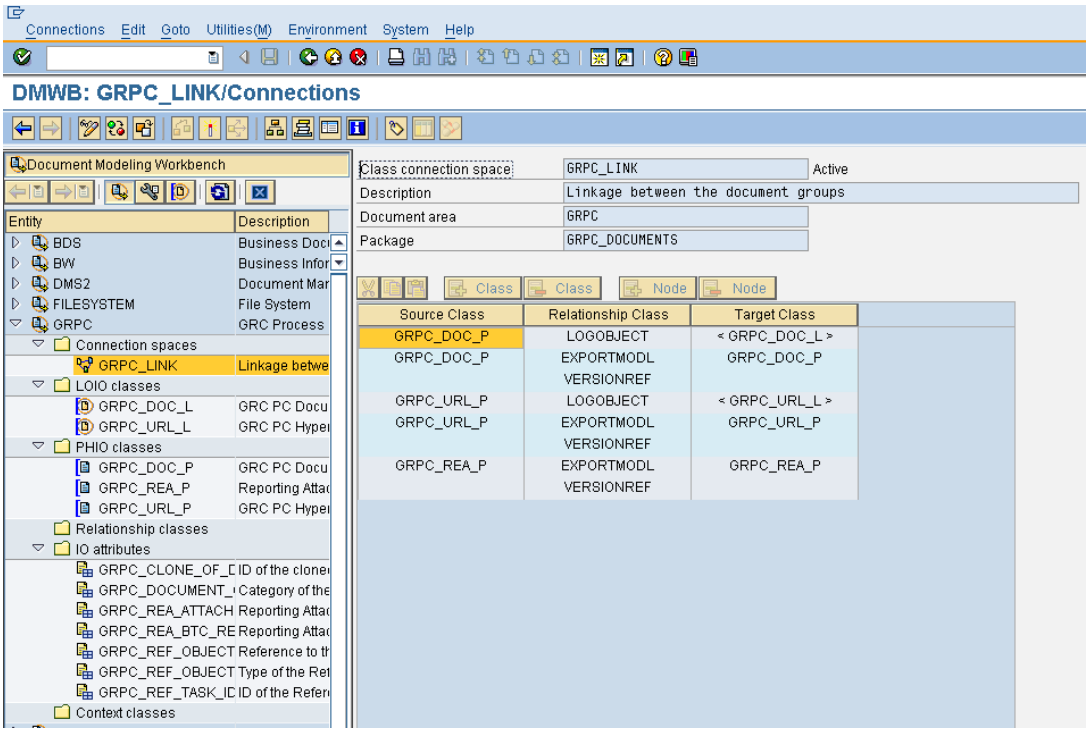
As you can see the only document class which doesn't have it's corresponding logical class is **GRPC_REA_P**, attachments of this class are documents holding output of reporting engine (printing reports (PDF file) or online reports results (OTF file)).

KPro Document Modeling Workbench – TA dmwb – is transaction that maintains so called document models – which is in fact set of document classes (logical and physical), their attributes and relations between them. The document model is named GRPC. See screenshots from TA ***dmwb*** below:

Main screen of TA *dmwb*:

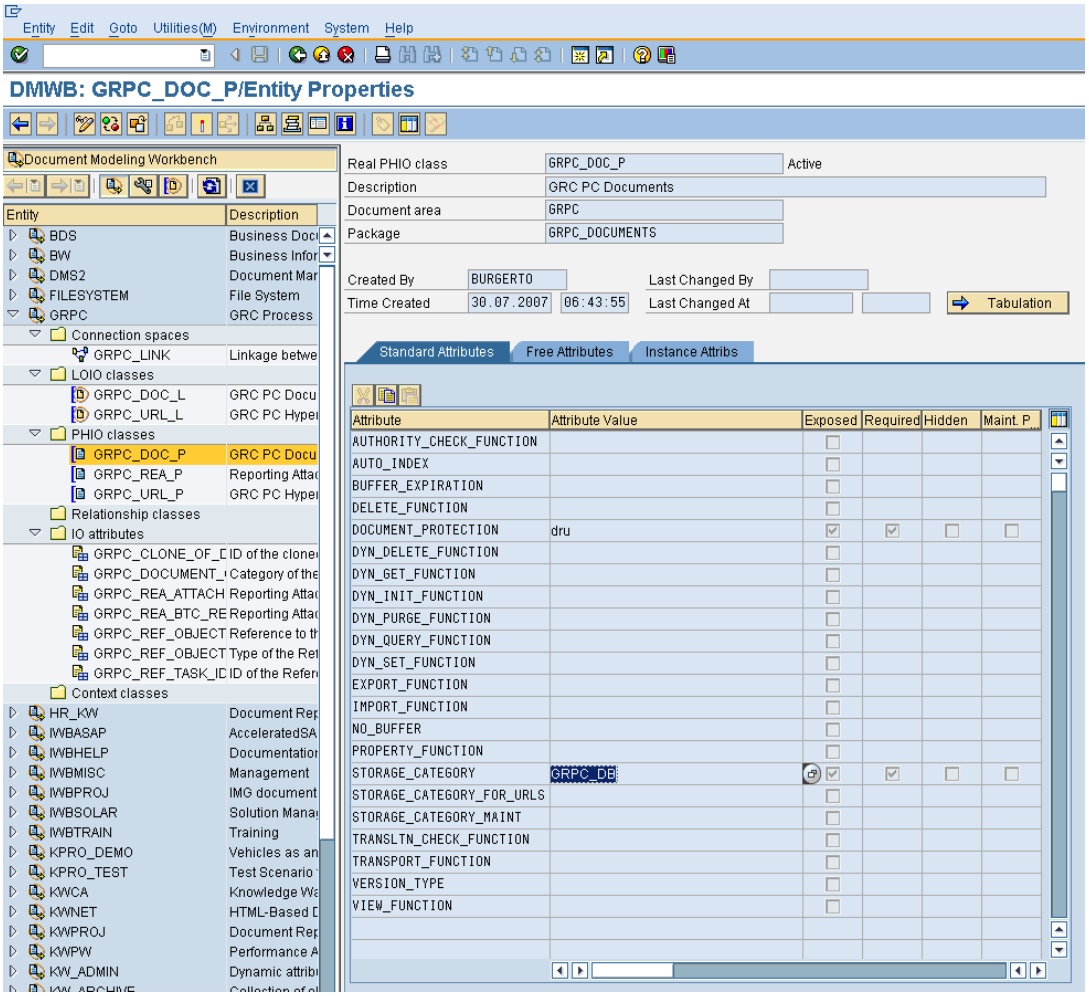


TA *dbwb* - Relations between document classes:

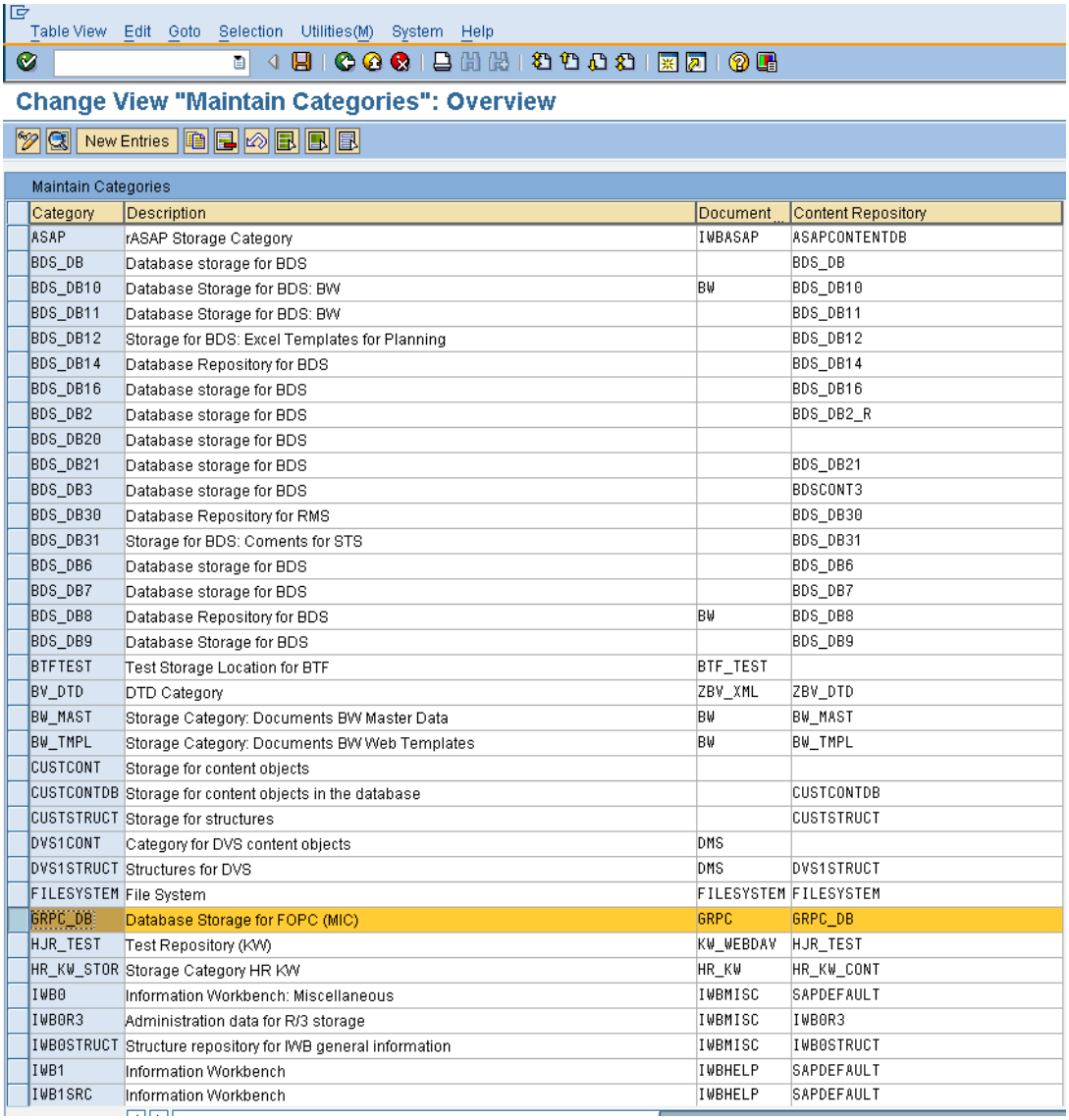


TA *dmwb* – Documents Modeling Workbench – GRPC model:

'STORAGE_CATEGORY' attribute pre-filed for process control document classes

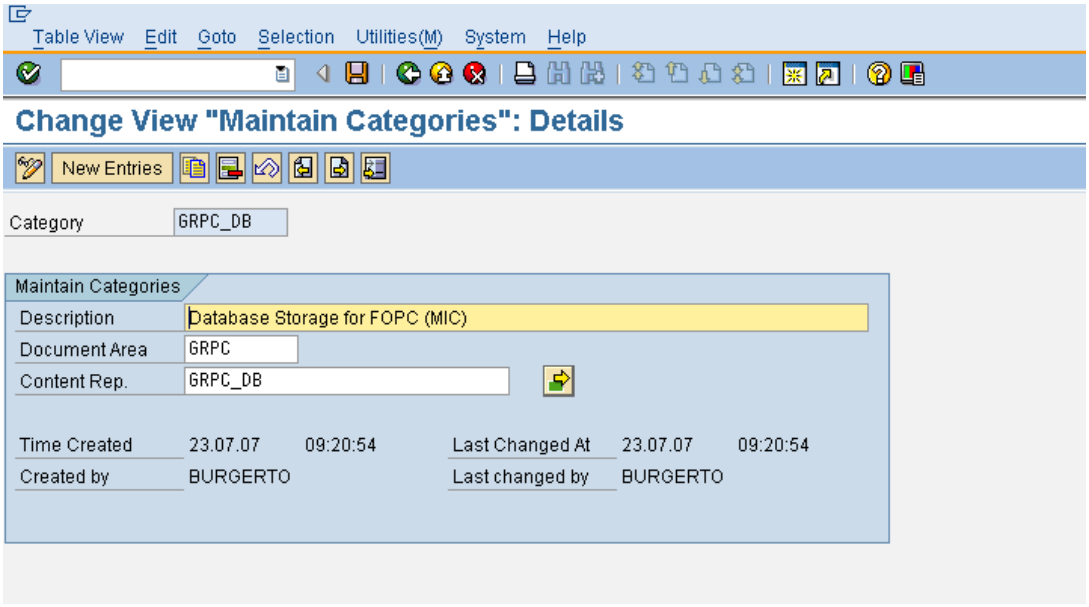


TA *oact* – KPro storage categories –category 'GRPC_DB':

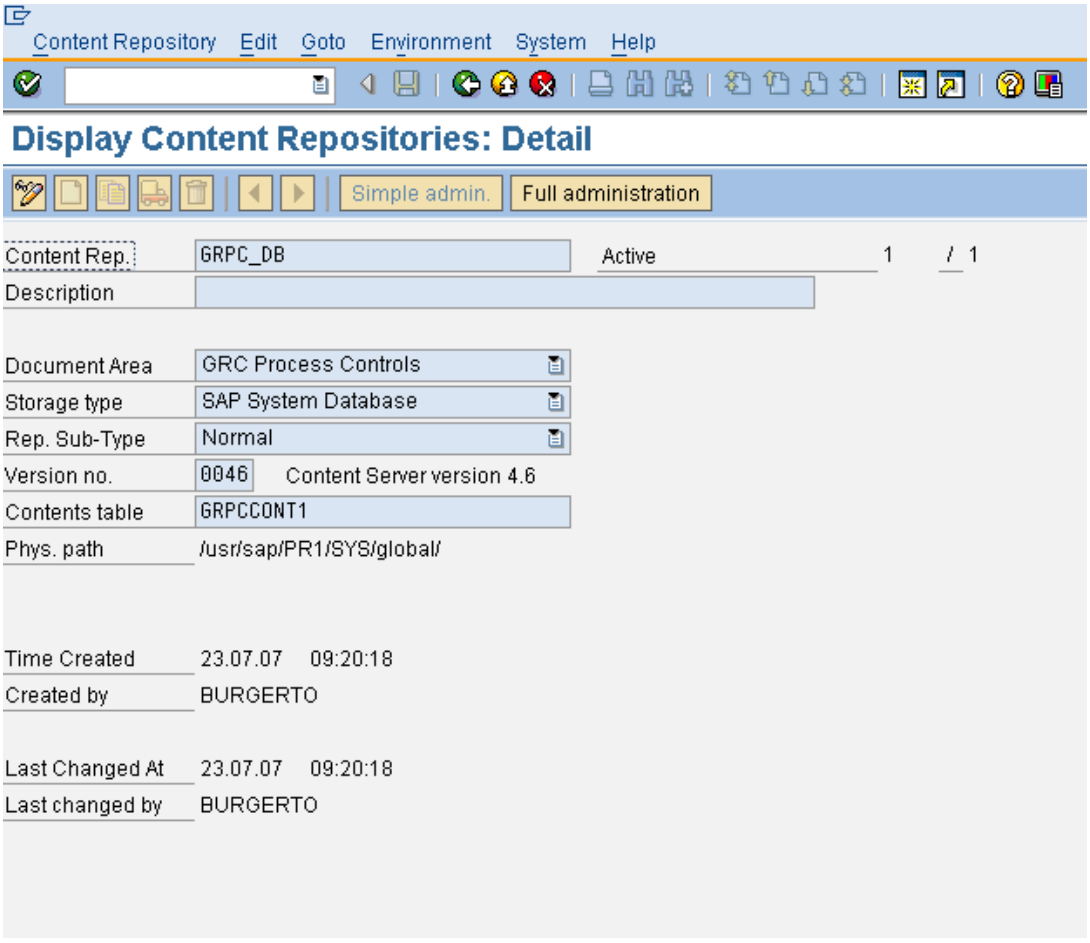


Category	Description	Document	Content Repository
ASAP	rASAP Storage Category	IWBASAP	ASAPCONTENTDB
BDS_DB	Database storage for BDS		BDS_DB
BDS_DB10	Database Storage for BDS: BW	BW	BDS_DB10
BDS_DB11	Database Storage for BDS: BW		BDS_DB11
BDS_DB12	Storage for BDS: Excel Templates for Planning		BDS_DB12
BDS_DB14	Database Repository for BDS		BDS_DB14
BDS_DB16	Database storage for BDS		BDS_DB16
BDS_DB2	Database storage for BDS		BDS_DB2_R
BDS_DB20	Database storage for BDS		
BDS_DB21	Database storage for BDS		BDS_DB21
BDS_DB3	Database storage for BDS		BDSCONT3
BDS_DB30	Database Repository for RMS		BDS_DB30
BDS_DB31	Storage for BDS: Coments for STS		BDS_DB31
BDS_DB6	Database storage for BDS		BDS_DB6
BDS_DB7	Database storage for BDS		BDS_DB7
BDS_DB8	Database Repository for BDS	BW	BDS_DB8
BDS_DB9	Database Storage for BDS		BDS_DB9
BTFTEST	Test Storage Location for BTF	BTF_TEST	
BV_DTD	DTD Category	ZBV_XML	ZBV_DTD
BW_MAST	Storage Category: Documents BW Master Data	BW	BW_MAST
BW_TMPL	Storage Category: Documents BW Web Templates	BW	BW_TMPL
CUSTCONT	Storage for content objects		
CUSTCONTDB	Storage for content objects in the database		CUSTCONTDB
CUSTSTRUCT	Storage for structures		CUSTSTRUCT
DVS1CONT	Category for DVS content objects	DMS	
DVS1STRUCT	Structures for DVS	DMS	DVS1STRUCT
FILESYSTEM	File System	FILESYSTEM	FILESYSTEM
GRPC_DB	Database Storage for FOPC (MIC)	GRPC	GRPC_DB
HJR_TEST	Test Repository (KW)	KW_WEBDAV	HJR_TEST
HR_KW_STOR	Storage Category HR KW	HR_KW	HR_KW_CONT
IWB0	Information Workbench: Miscellaneous	IWBMISC	SAPDEFAULT
IWB0R3	Administration data for R/3 storage	IWBMISC	IWB0R3
IWB0STRUCT	Structure repository for IWB general information	IWBMISC	IWB0STRUCT
IWB1	Information Workbench	IWBHELP	SAPDEFAULT
IWB1SRC	Information Workbench	IWBHELP	SAPDEFAULT

TA *oact* – detail of storage category 'GRPC_DB' – is mapped to 'GRPC_DB' repository:



TA *oac0* – detail of storage repository 'GRPC_DB' - based on DB table GRPCCONT1:



4. Configuring Content Server Alias

In complex environments, especially those with firewalls, multiple servers may be involved in the handling of a request, regardless of the locations of the servers. Each of these servers plays a role in retrieving the requested content, or, as the case may be, in forwarding the request (similarly to cascaded caches).

These servers are known as **content server aliases** (in the sense that they "represent" the content server).

Configuring a Content Server Alias

You have to make the following Customizing settings:

- SCMSCSPX – content server aliases (the CSPX here comes from content server proxies, an alternative term for aliases).

  The technical data of the alias is stored in this table. By specifying the technical data of the content server, you define which content server is represented. Ensure that you enter all the technical details correctly, including upper-case and lower-case characters. The individual fields have the following meanings:

PX_SERV	Host name of the alias server
PX_PORT	Port of the alias server
PX_SPORT	SSL port of the alias server
PX_SCRPT	HTTP script of the alias server
CS_SERV	Host name of the content server
CS_PORT	Port of the content server
CS_SPORT	SSL port of the content server
CS_SCRPT	HTTP script of the content server
NO_GET	Alias is not used for ‘cacheable’ get requests
INACTIVE	Entry inactive

- SCMSCSPL – other locations of content server aliases (the CSPL here comes from content server proxy locations).

This table contains other locations, besides those entered in table SCMSHOST that can also be used for the alias server.

The data for the content server must be exactly the same as that in the Customizing for the repository (transaction OAC0).

- Only from release 4.6D can you explicitly specify the port. Up to release 4.6C, the port is added at the end of the content server name in the form `:<port>`.
- You have to explicitly define the HTTP port when defining the alias server. In releases before 4.6D, leave the SSL port at its initial value.
- The field NO_GET can be used to specify that a content server alias is not to be used for 'cacheable' get requests.

This is useful if the caches can support the cacheable get requests better than the alias.

The field INACTIVE can be used to stop the alias in question being used.

Determining the Alias

- The system checks whether there is an alias for the content server in question at the client's location.
- If there is no alias at that location, the system looks for another alias that can be used.
- If an alias is found, the technical data of that alias is used instead of those of the content server.
- If more than one alias is found, load distribution is used automatically.

Multi-layer caching and aliases can also be used in combination. The system always looks for an alias first. If it finds one, the technical data of the alias is used instead of the data of the content server. This means that the location of the server may change. This information is then taken into account when the cache server is being located.

Constraints

Caching and content server aliases are only used if the client location is known when the URL is being constructed. Usually, the client location is known if the Knowledge Provider processes the URL. If, on the other hand, the URL was requested by another application, and the Knowledge Provider does not know where the URL is going to be used, the system cannot find out the location of the client. In this case, the URL that points directly to the content server is always returned. The caches and the alias server do not, therefore, play any role.

This can often be the case with ArchiveLink.

For more information please refer Appendix A.

5. Process control2.5 Attachments Attributes

KPro Document Attributes – sometimes also known as ***document property*** - simply place holder within document storage system where some additional information about the document can be stored. There exists several standard document attributes (like creation date/time, document description and many other technical attributes). These standard attributes, if set, may have influence on document storage and retrieval (i.e. attribute STORAGE_CATEGORY). More about standard attributes can be found in KPro documentation resources.

Documents in KPro may have also application specific attributes defined by developer. They distinguish between ***navigational attributes*** and ***vertical attributes***. Those navigational attributes are stored in DB tables in predefined columns (defined by developer) and thus are more selective then vertical attributes***,*** which are stored in separate table.

The ***main advantage of navigational attributes*** is that they are stored in special columns of DB table of documents. Thus ***access to them is much faster*** then access to vertical attributes. Thus they are ideal for storing secondary keys of document (like Process control Object ID, document category, etc.). And this is what Process control attachments backend does -> it stores so called ***application object key*** in 3 (actually in 2) navigational attributes. This allows to quickly retrieving all documents which are attached to given Process control object.

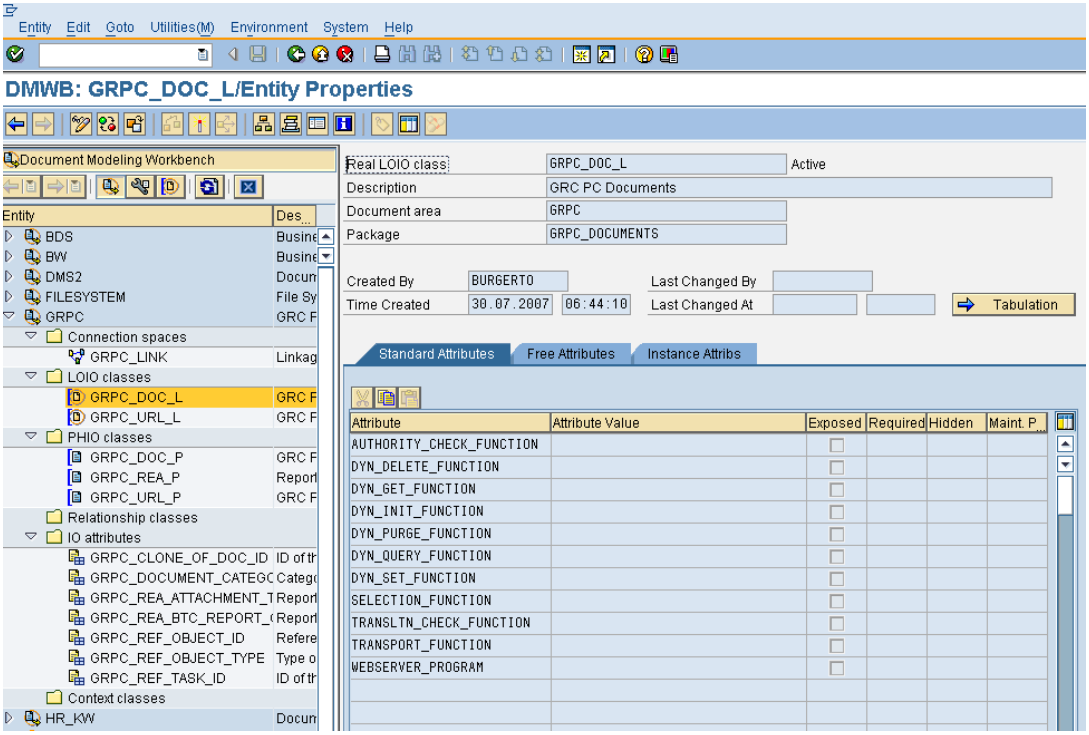
List of all the attributes defined in GRPC documents model:

Kpro document attribute	Description
GRPC_CLONE_OF_DOC_ID	GUID of document from which was current document cloned. This attribute is added to attachment if it takes part during some document cloning (which may happen (if it is customized in IMG) during carry forward of some assessment/testing/issue or remediation plan. During carry forward are existing attachments (documents and links) copied to new Process control object. Of course once attachment is cloned the it doesn't change its KPro document class.
GRPC_DOCUMENT_CATEGORY	Document category. This attribute was invented during later releases of CGVMIC (SP11) to have more fine

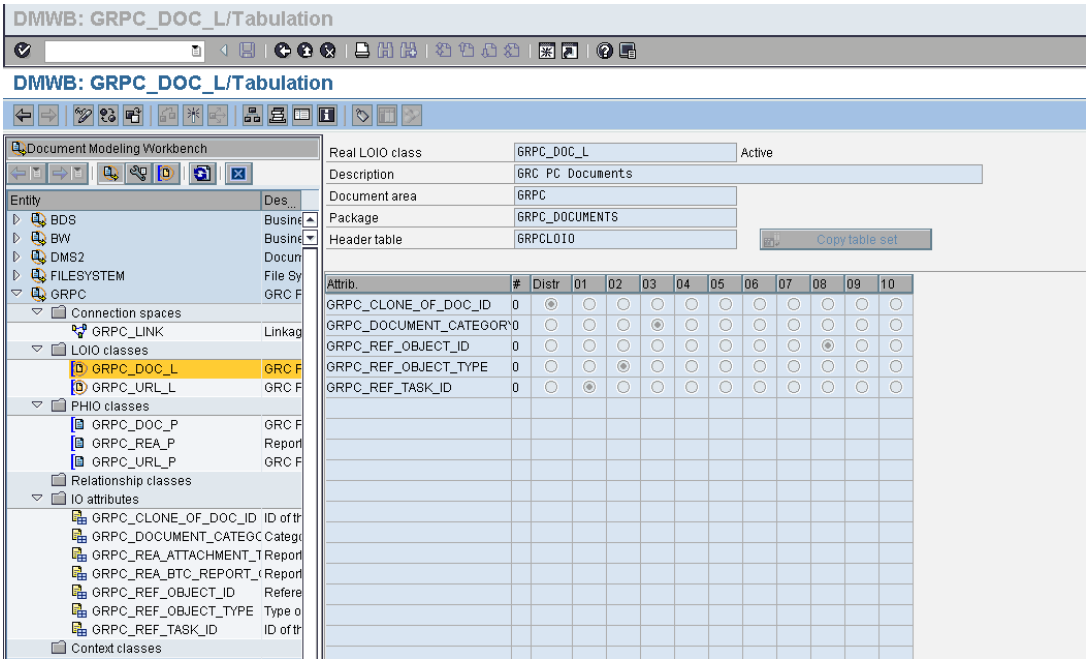
	differentiation between documents belonging to one Process control object – i.e. testing procedure documents of control and normal documents of control. The document category is kind of attachment type classification. You can find corresponding values in DDIC domain grpc_document_category and constants in type-pool constants structure grpcd_c_document_category.
GRPC_REA_ATTACHMENT_TYPE	Type of Background report attachment. This attribute is added to reporting attachments (document class GRPC_REA_P) and indicates the type of the attachment – corresponding values are stored in DDIC domain grpc_btc_attachment_type and constants may be found in GRPCR type-pool in constants structure grpcr_c_btc_report_attachment. Currently only PDF and OTF attachment types are implemented, the rest was reserved for future. In advance each background report (if not yet deleted from system) has record in DB table GRPC_BTC_REPORTS where you can also find the attachment type (if any report attachment exists). So this attribute redundant
GRPC_REA_BTC_REPORT_GUID	GUID of owning background report. This attribute is added to reporting attachments (document class GRPC_REA_P) and holds the ID of background report to which the document belongs. In opposite to other attachments in Process control, the reporting attachments are not looked up the way as other Process control objects (detail in chapter below), instead each background report knows which reporting attachments is belonging to it. So again, this attribute is redundant information for debugging and tracking purposes.
GRPC_REF_OBJECT_ID	Referencing object identifier (among sub object types). This attribute

	together with attribute GRPC_REF_OBJECT_TYPE is supposed to be the key to the Process control object to which the attachment belongs to. Details in chapter below.
GRPC_REF_OBJECT_TYPE	Referencing object type. This attribute together with attribute GRPC_REF_OBJECT_ID is supposed to be the key to the Process control object to which the attachment belongs to. Details in chapter below.
GRPC_REF_TASK_ID	This attribute was also part of the Process control Object key. However this didn't allow sharing attachments of one Process control object among several Process control tasks, so this attribute was excluded from Process control Object Key. Current meaning of the attribute is that it holds ID of the Process control Task in which the document was 1st time created. There is no business logic defined on this attribute.

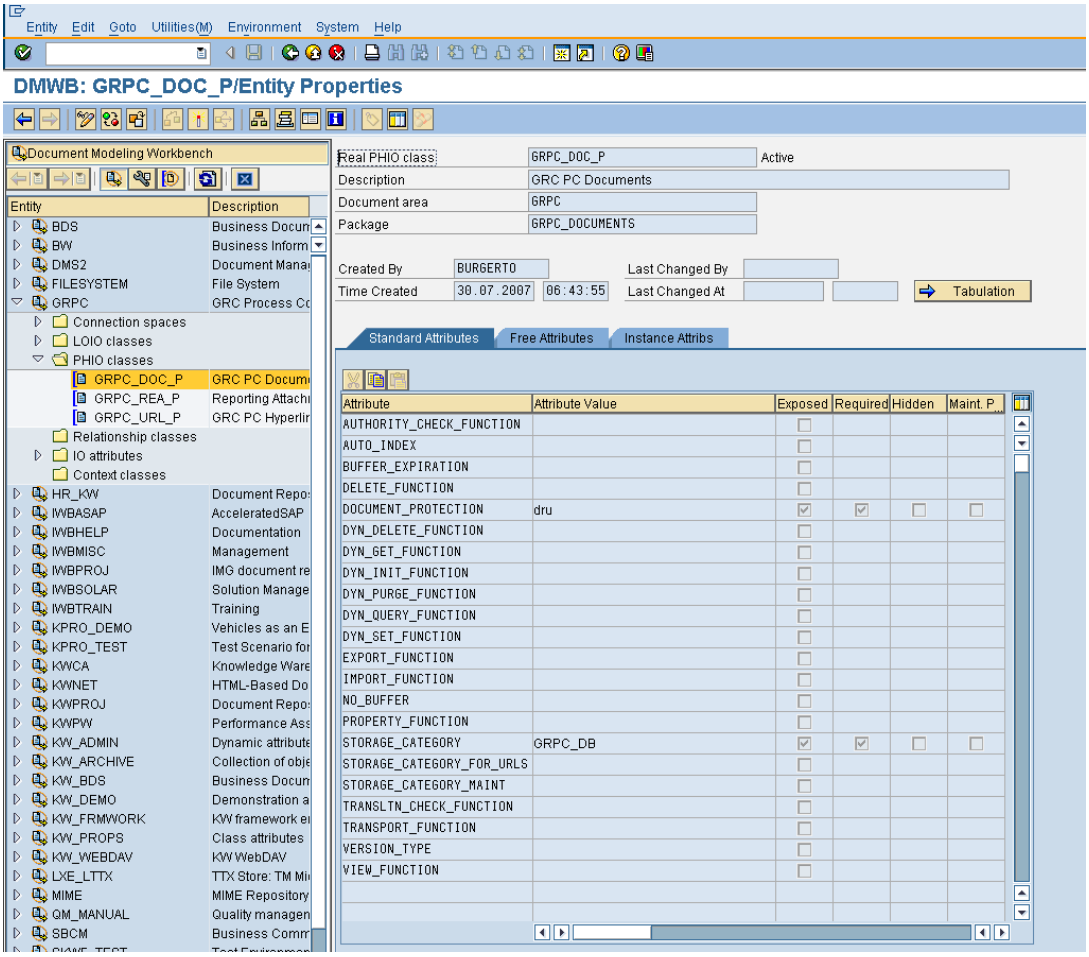
TA *dmwb* –attributes of GRPC_DOC_L document class:



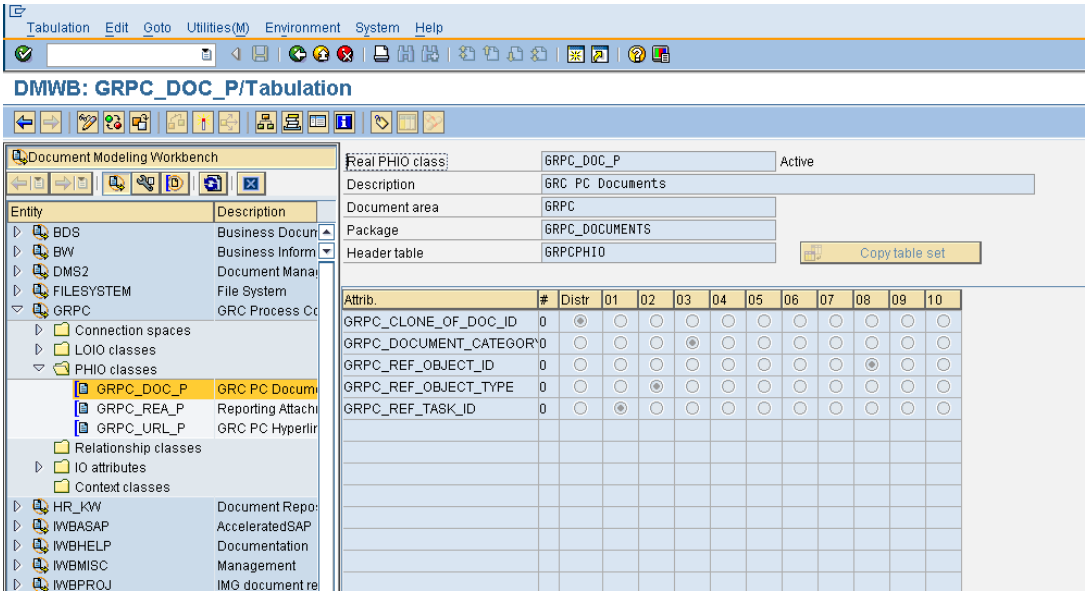
TA *dmwb* – GRPC_DOC_L document class – mapping of navigational attributes:



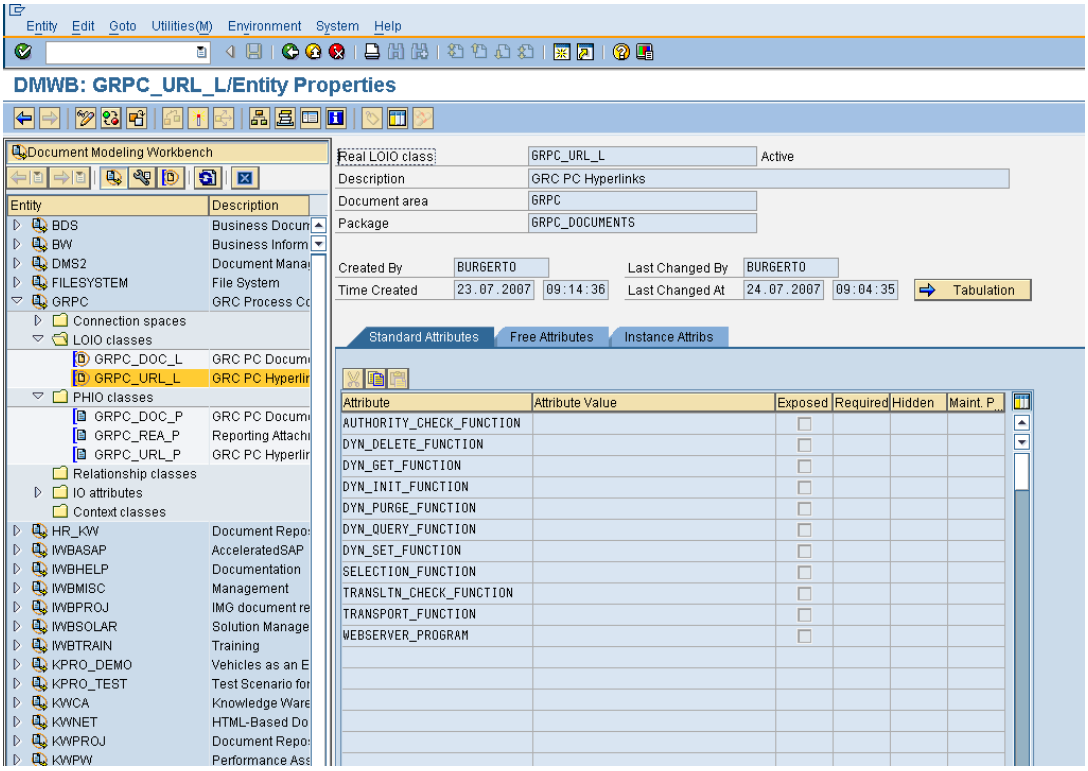
TA *dmwb* –attributes of GRPC_DOC_P document class:



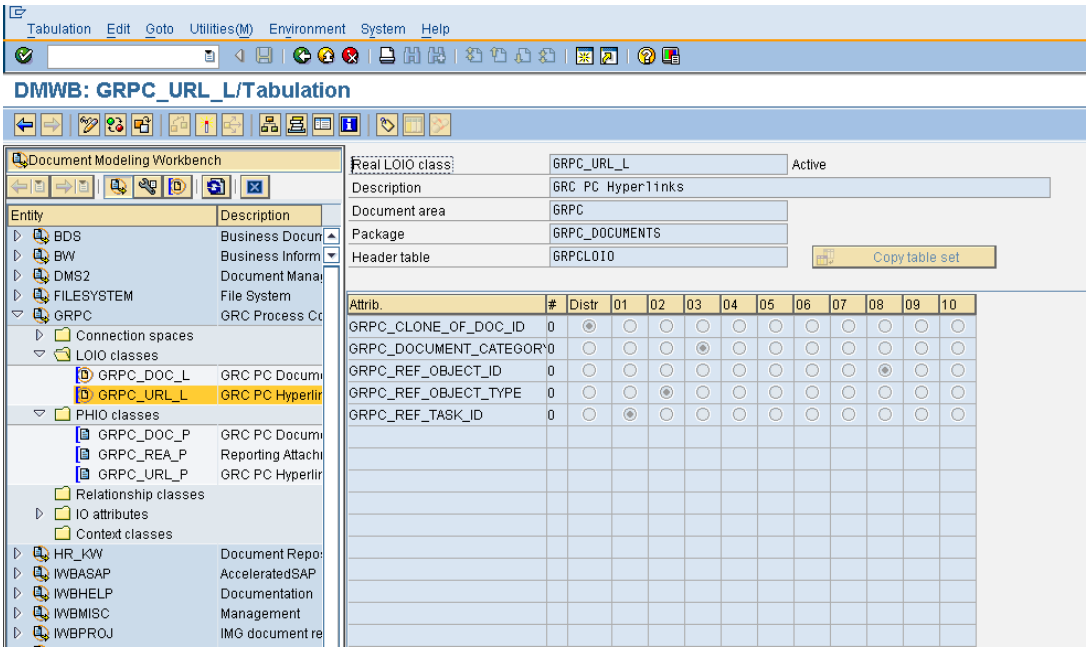
TA *dmwb* – GRPC_DOC_P document class – mapping of navigational attributes:



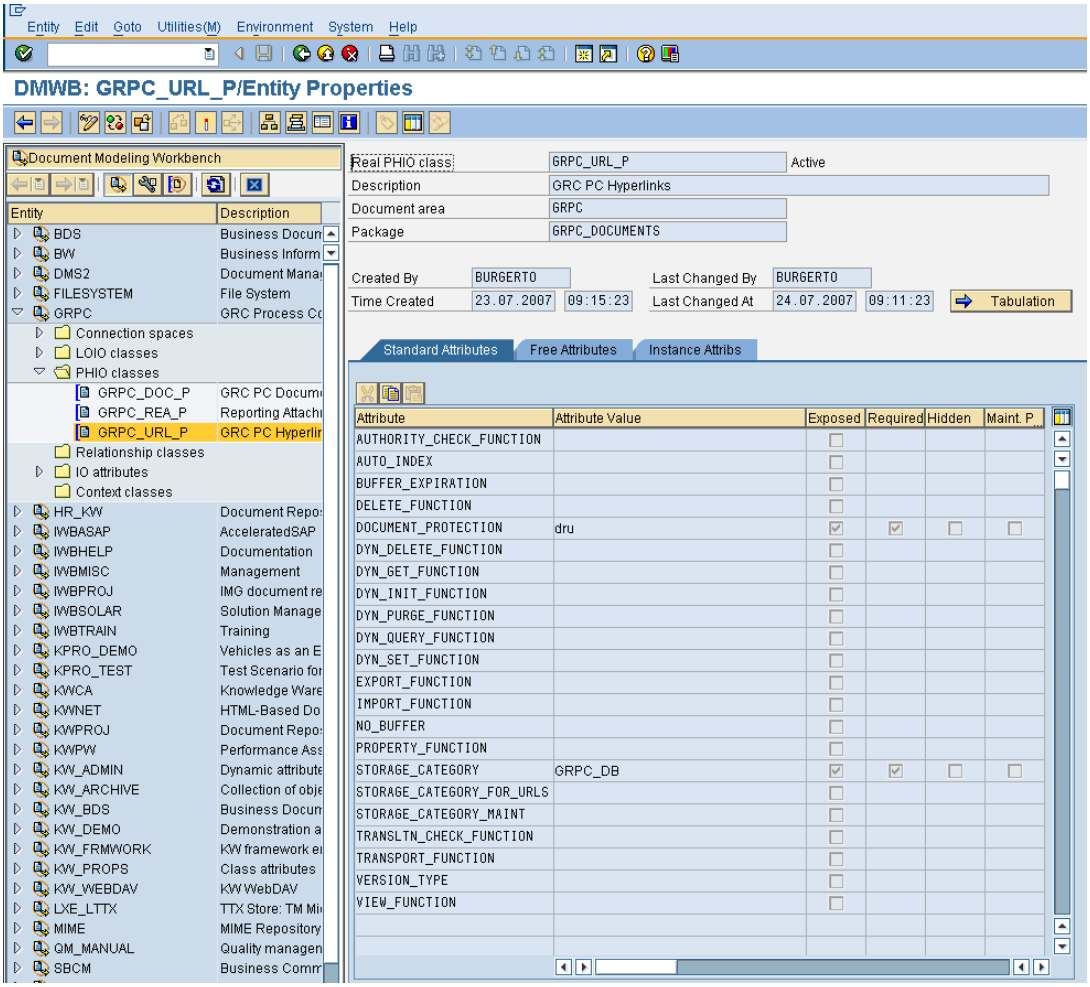
TA *dmwb* –attributes of GRPC_URL_L document class:



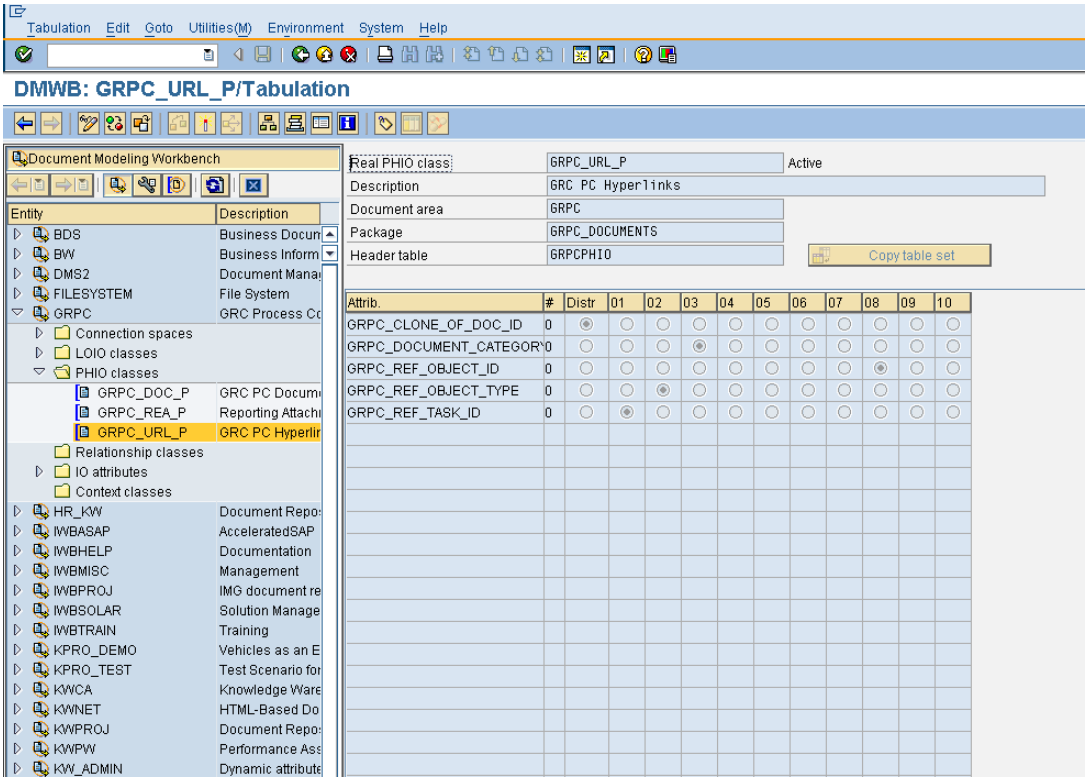
TA *dmwb* – GRPC_URL_L document class – mapping to navigational attributes:



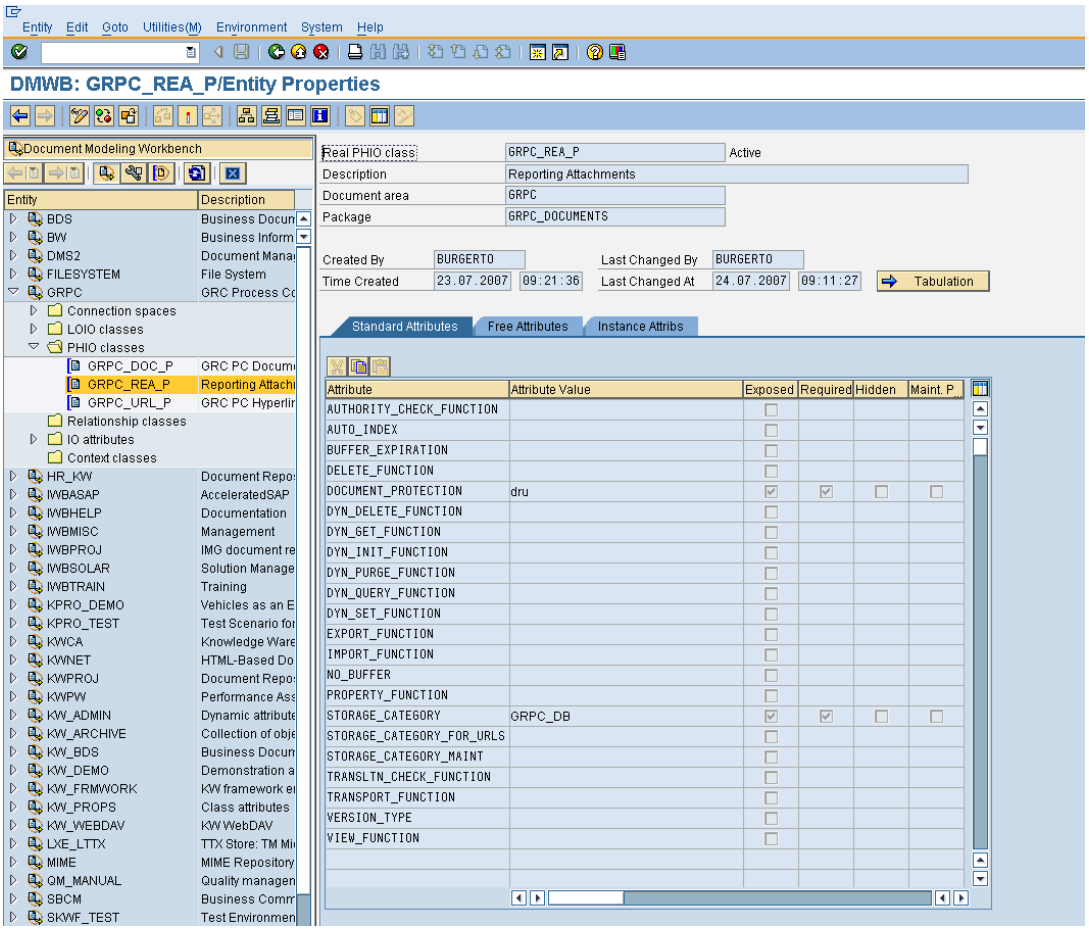
TA *dmwb* –attributes of GRPC_URL_P document class:



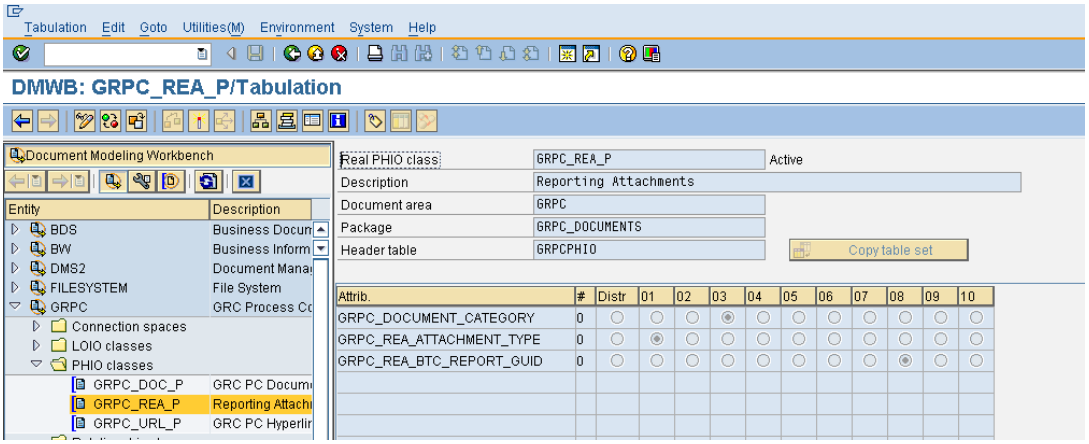
TA *dmwb* – GRPC_URL_P document class – mapping to navigational attributes:



TA *dmwb* –attributes of GRPC_REA_P document class:



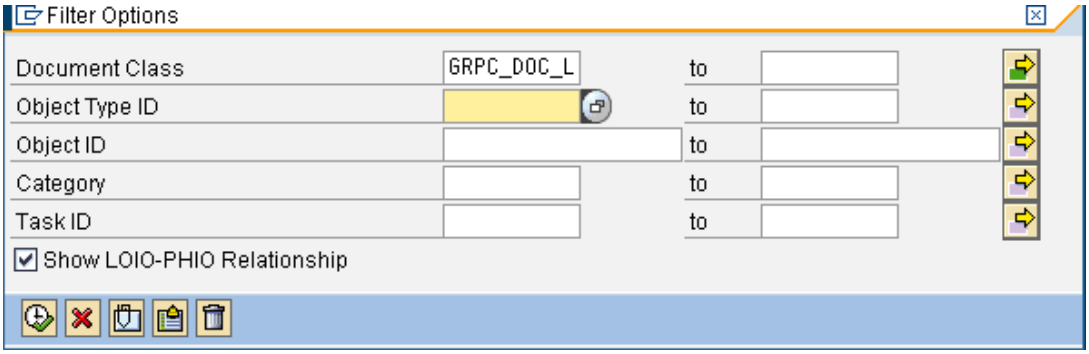
TA *dmwb* – GRPC_REA_P document class – mapping to navigational attributes:



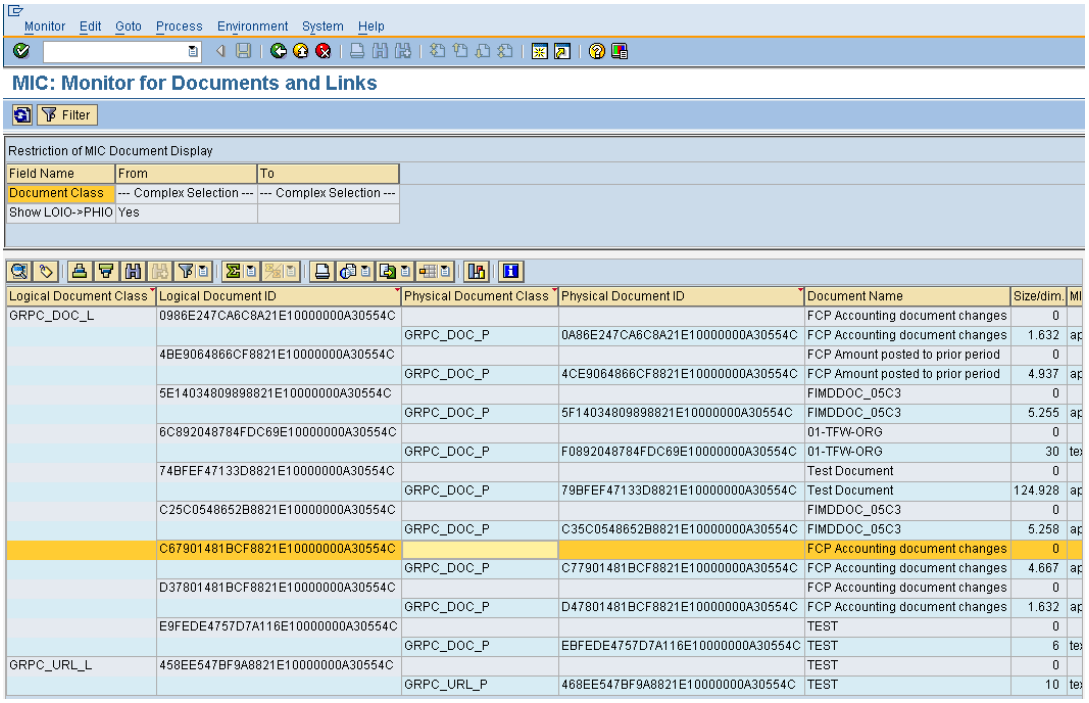
6. Documents monitor – TA *grpc_doc_monitor*

Since several releases there is Process control attachments monitor GUI transaction - TA ***grpc_doc_monitor*** - which can be used for lookup for existing attachments during debugging some functionality or checking existence of some documents/links. Its main features are lookup for attachments by given parameters on selection screen (Document class – LOIO, PHIO, Task ID, PROCESS CONTROL Object ID, PROCESS CONTROL Object type, Document Category). All Process control attachments which fulfills given selection criteria are then listed in ALV grid and user is allowed to see all attributes of logical/physical documents, all PHIOs which belongs to one LOIO (in case of more versions of the attachment). There is also hidden debug menu – ok-code 'debug_menu' – which allows some advanced (from security point of view also problematic) features – like i.e. generation of download URL for given attachment in the detail view of attachments attributes.

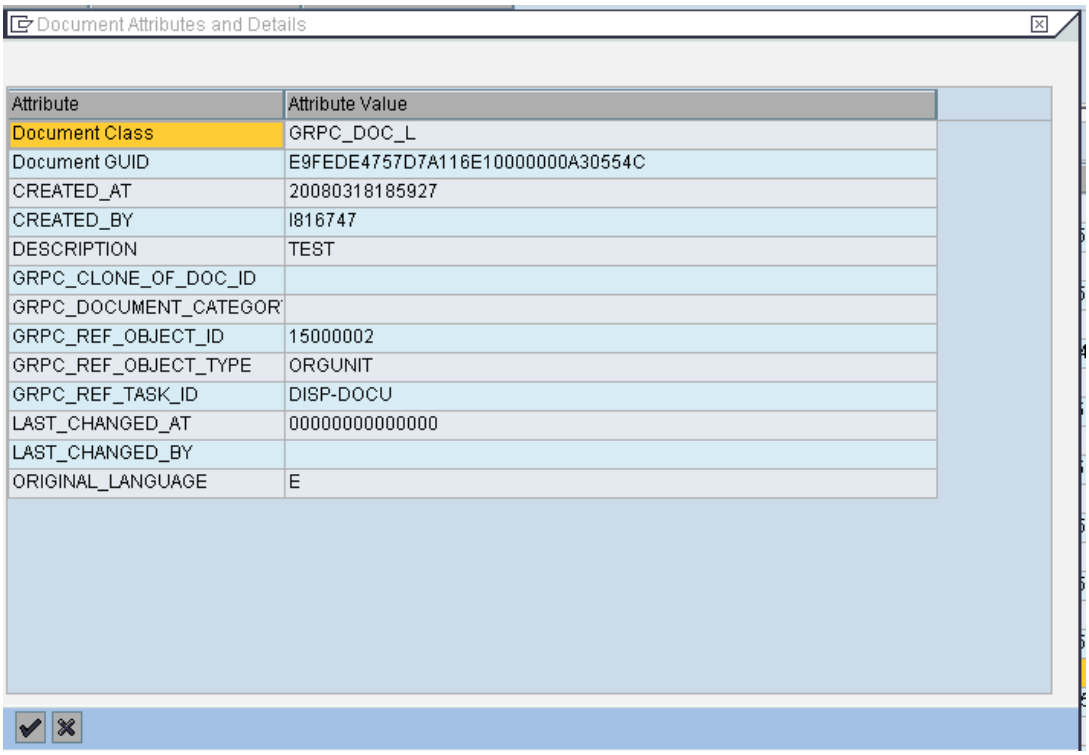
Selection screen



TA *grpc_doc_monitor* - ALV grid with listed Process control attachments:

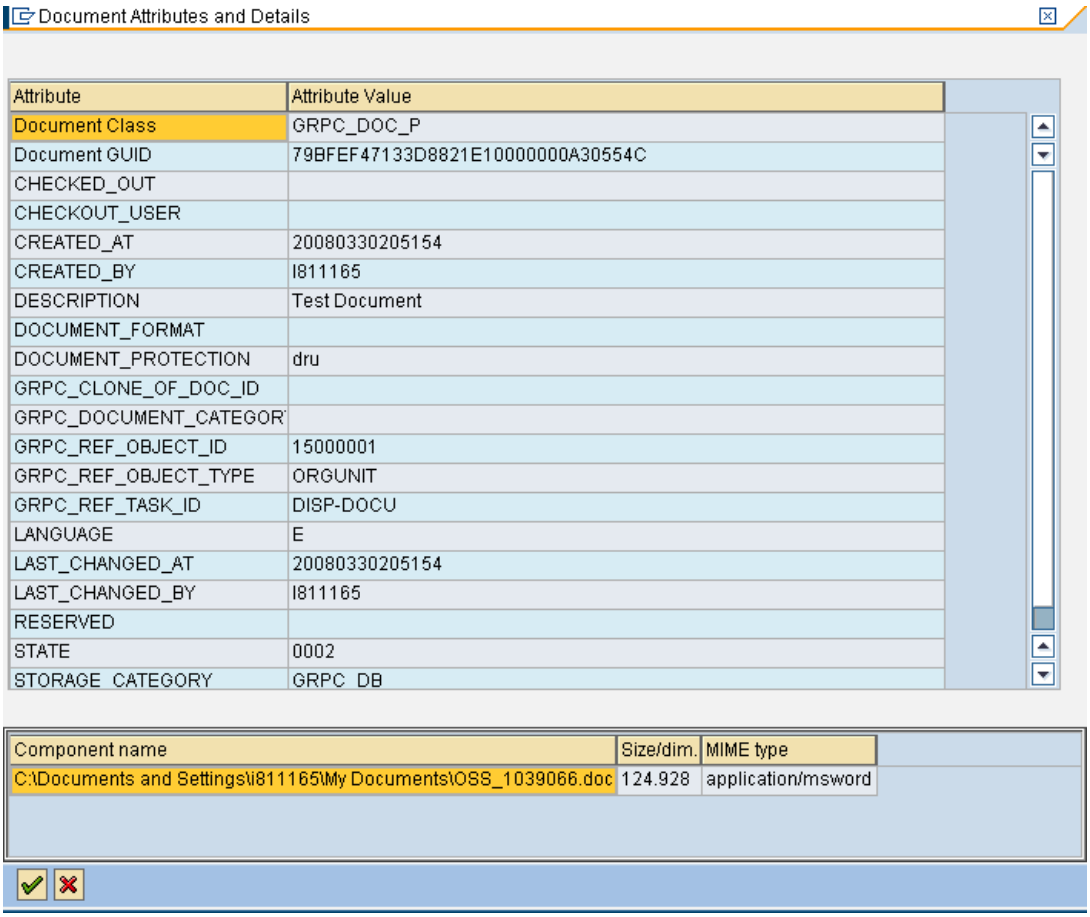


TA *grpc_doc_monitor* – details of logical document of Process control attachment:



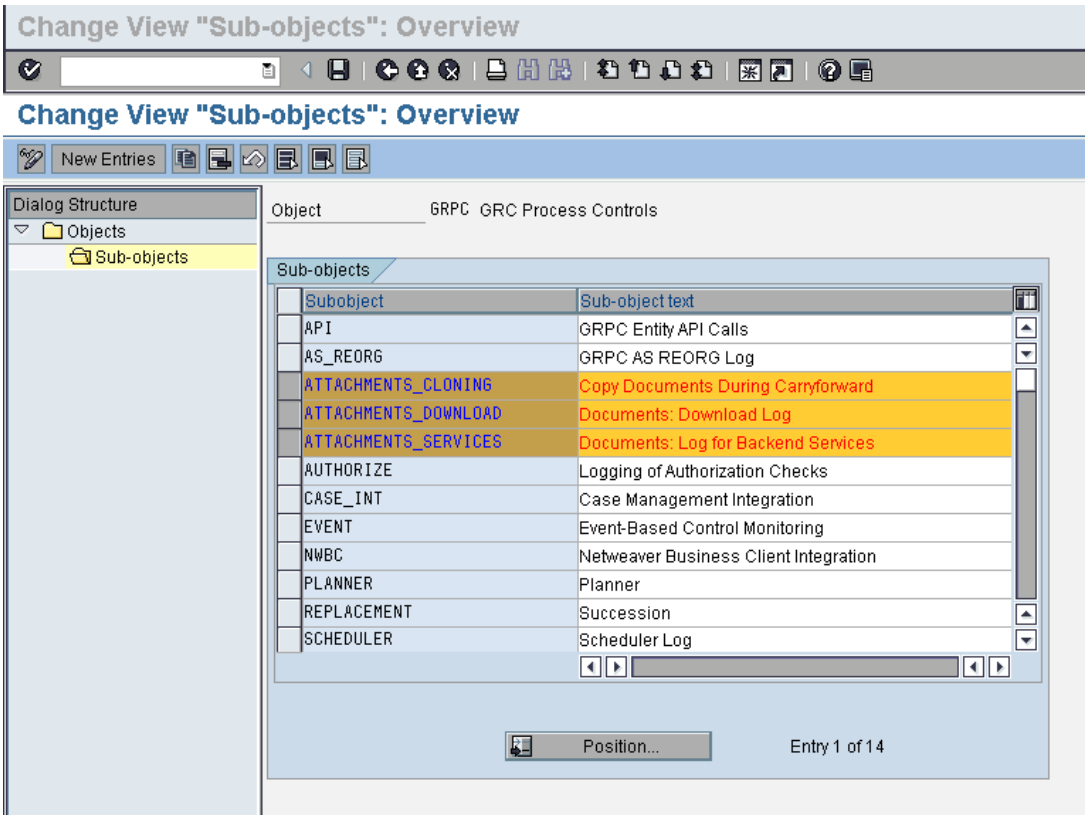
TA *grpc_doc_monitor* – details of physical document of Process control attachment:

Including generated URL for download enabled via hidden Debug menu



TA *slg0* – definition of SAP application log objects and log sub-objects:

Attachments relevant sub-objects of GRPC log object are marked



TA *slg1* – Example of content of an PROCESS CONTROL SAP application log content:

Using this transaction code we can see object GRPC sub-object ATTACHMENTS_SERVICES where deletion of an attachment is recorded.

7. GRPC_DOCUMENTS development package

GRPC_DOCUMENTS development package holds all the ABAP programs, classes and function modules responsible for UI and backend functionality of PROCESS CONTROL attachments.

7.1 Important DDIC Objects

7.1.1 DB tables

GRPCC*, GRPCL*, GRPCP*

- DB tables generated by TA ***dmwb*** from documents model GRPC. They are used to store directory information about each Process control Attachment (including properties and their values)
- **GRPCCONT1** is default content repository table for Process control attachments (for storing of content of PHIO documents). But this can be overridden by TA ***oac0*** and TA ***oact***.

GRPCDOCDOWNLCFG

- Holds information for which document class is Process control Alternative Downloader activated and with which parameter settings.

GRPC_DOC_TICKETS

- Stores security tickets for documents downloaded by Process control Alternative downloader (if switched on and configured for secured transfer).

- Once any part of Process control application (UI, backend) requests a download URL for document of class for which PROCESS CONTROL Alternative Downloader is switched on with security flag set (via method ***CL_GRPC_DOCUMENTS_MANAGER-> GET_DOCUMENT_DOWNLOAD_URL***) then the security ticket with proper parameters is created for the download operation and the ticket ID is encoded into returned download URL.

- As the document is requested for download via URL served by Process control Alternative Downloader, the ticket encoded in the download URL is checked for validity 1st and if it's okay, then document content is returned in HTTP response.

- Non valid (outdated) tickets are cleaned automatically during download process and/or can be also cleaned if ABAP report ***GRPC_DOCUMENTS_TICKET_CLEANER*** is scheduled for periodical execution.

7.1.2 DDIC structures

GRPC_BTC_JOB_INFO

- This structure is by accident in this package, it belongs to GRPC_REPORTING package (background reporting backend)

7.1.3 Lock objects

- **E_GRPCDOCTICKET**
  - Synchronization object for accessing DB table GRPC_DOC_TICKETS
- **E_GRPCMIGRATION**

- Synchronization for Process control HR-Links migration tasks (if more then one started in parallel)

7.2 ABAP Programs

➢ **GRPC_CLONE_CASE_DOCUMENTS**

- Implements the attachments cloning functionality. Should not be called directly, instead it is used by other API functions to ensure attachments cloning. The execution protocol is stored in SAP application log – TA slg1 – object GRPC, sub-object ATTACHMENTS_CLONING.

➢ **GRPC_DOCUMENTS_CLONING_JOB**

- ABAP report to be planned periodically for ensuring Attachments cloning (mentioned in IMG). After a CASE cloning is finished (which stores cloned cases into DB table GRPCCASECLONE) a system event – event name = 'SAP_END_OF_JOB', event parameter = 'PROCESS CONTROL_CARRY_FORWARD_DOC') - is fired by PROCESS CONTROL CASE management sub-system. The planned background job (this ABAP report - registered for this event) is then started and it reads all cloned CASE objects from the DB table and ensures cloning of attachments for those cloned PROCESS CONTROL CASE objects – by calling ABAP report GRPC_CLONE_CASE_DOCUMENTS with corresponding parameters.

➢ **GRPC_DOCUMENTS_MONITOR**

- PROCESS CONTROL attachments lookup report. Allows to find out PROCESS CONTROL attachments according to given search criteria (document class, object ID, task ID, object type). Allows also using pattern search. Once found documents are read from KPro, you can then inspect all the properties of given PROCESS CONTROL attachment.
- Hidden ok-code "debug_menu" allows switch on/off enhanced functionality:
  - download tickets of PROCESS CONTROL alternative downloader
    - inspection
    - deletion of outdated tickets
    - deletion of all tickets
  - download URL generation (in SAP GUI)

➢ **GRPC_DOCUMENTS_MONITOR_FILTER**

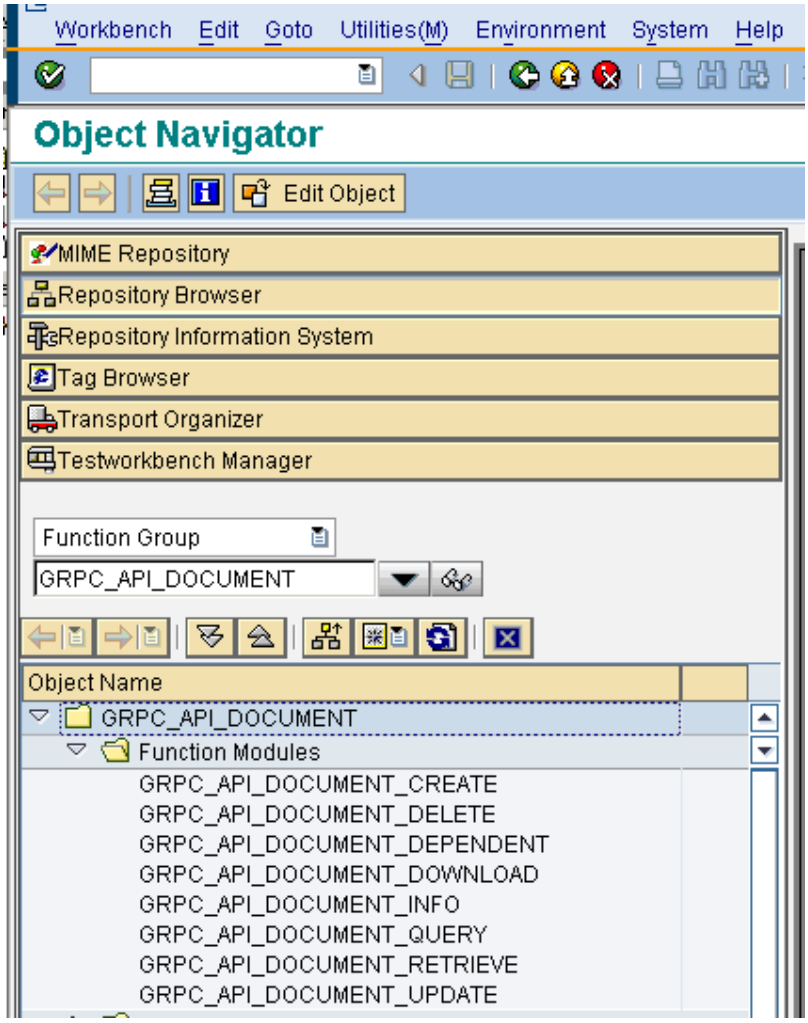
- Support ABAP report for GRPC_DOCUMENTS_MONITOR. Implements only selection screen for filtering options of PROCESS CONTROL attachments.

➢ **GRPC_DOCUMENTS_TICKET_CLEANER**

- Implements cleaning of all or outdated download tickets of PROCESS CONTROL alternative downloader. The execution protocol is stored in SAP application log – TA slg1 – object GRPC, sub-object ATTACHMENTS_DOWNLOAD.

7.3 Important Function Modules

The following standard APIs can be viewed from transaction SE80, under the function group GRPC_API_DOCUMENT



8. Other important development objects

- **Message Class - GRPCD**

  - Hold all messages of attachments sub-system in PROCESS CONTROL, including UI messages and technical messages. There are some logical sections in the message class, once creating new message, try to put it into correct section.

- **Type-pool GRPCD**

  - This type pool holds most of structures and constants defined and used in PROCESS CONTROL attachments sub-system.

9. Hints for debugging

9.1 Important transaction codes for debugging Process Control 2.5 attachments

- ***Dmwb*** – KPro Document Modeling Workbench
- ***Oac0, oact*** – KPro customizing of documents Storage Categories and Repositories
- ***Skpr08*** - KPro – overriding storage category per document class
- ***Skpr07*** - KPro Monitoring
- ***Spro*** **-** Process control IMG
- ***grpc_doc_monitor*** - Process control Documents Monitor
- ***Slg0, slg1*** **-** *SAP Application Log*

10. Transport of Documents/Attachments/URL

The documents and the URL get stored in the content server as the plain text file. The link between the PC2.5 and the documents are application specific which is defined in PC2.5 application.

So if we want every time we transport the PC2.5 objects from one landscape to other we have to re upload the attachments.

Please refer Appendix B for additional information.

11. Appendix

Appendix A – SAP notes related to caching and content server alias configuration

Please refer the following notes for additional information on caching and content server alias configuration.

- 0181696, Caching
- 0209478, SAP KPro Server Infrastructure Components 4.6C
- 0303278, SAP KPro Server Infrastructure Components 4.6D
- 0352518, Using the SAP Content Server Cache
- 0376033, Cache Server Knowledge Warehouse 5.1
- 0407520, Information on the Cache Server

Appendix B – SAP notes related to to document transport

Please refer the following notes for additional information on transport of DMS

- 868385, Transport document templates to another system
- 602820,Transporting characteristics that are document properties
- 858355,Multiple inclusions in different transport requests

12. Comments and Feedback

Both comments and feedback are very welcome. Please send them to:

- Debraj Roy RIG Process Control debraj.roy @sap.com
- Raj Behera RIG Manager raj.behera@sap.com