

SAP NetWeaver '04



SAP<sup>®</sup> Exchange  
Infrastructure 3.0:  
Best Practices for  
Naming Conventions

Document Version 1.0 – July 3, 2006



SAP AG

Neurottstraße 16  
69190 Walldorf  
Germany  
T +49/18 05/34 34 24  
F +49/18 05/34 34 20  
[www.sap.com](http://www.sap.com)

© Copyright 2006 SAP AG. All rights reserved.

No part of this publication may be reproduced or transmitted in any form or for any purpose without the express permission of SAP AG. The information contained herein may be changed without prior notice.

Some software products marketed by SAP AG and its distributors contain proprietary software components of other software vendors.

Microsoft, Windows, Outlook, and PowerPoint are registered trademarks of Microsoft Corporation.

IBM, DB2, DB2 Universal Database, OS/2, Parallel Sysplex, MVS/ESA, AIX, S/390, AS/400, OS/390, OS/400, iSeries, pSeries, xSeries, zSeries, z/OS, AFP, Intelligent Miner, WebSphere, Netfinity, Tivoli, and Informix are trademarks or registered trademarks of IBM Corporation in the United States and/or other countries.

Oracle is a registered trademark of Oracle Corporation. UNIX, X/Open, OSF/1, and Motif are registered trademarks of the Open Group.

Citrix, ICA, Program Neighborhood, MetaFrame, WinFrame, VideoFrame, and MultiWin are trademarks or registered trademarks of Citrix Systems, Inc.

HTML, XML, XHTML and W3C are trademarks or registered trademarks of W3C®, World Wide Web Consortium, Massachusetts Institute of Technology.

Java is a registered trademark of Sun Microsystems, Inc. JavaScript is a registered trademark of Sun Microsystems, Inc., used under license for technology invented and implemented by Netscape.

MaxDB is a trademark of MySQL AB, Sweden.

SAP, R/3, mySAP, mySAP.com, xApps, xApp, and other SAP products and services mentioned herein as well as their respective logos are trademarks or registered trademarks of SAP AG in Germany and in several other countries all over the world. All other product and service names mentioned are the trademarks of their respective companies. Data contained in

this document serves informational purposes only. National product specifications may vary.

These materials are subject to change without notice. These materials are provided by SAP AG and its affiliated companies ("SAP Group") for informational purposes only, without representation or warranty of any kind, and SAP Group shall not be liable for errors or omissions with respect to the materials. The only warranties for SAP Group products and services are those that are set forth in the express warranty statements accompanying such products and services, if any. Nothing herein should be construed as constituting an additional warranty.

These materials are provided "as is" without a warranty of any kind, either express or implied, including but not limited to, the implied warranties of merchantability, fitness for a particular purpose, or non-infringement.

SAP shall not be liable for damages of any kind including without limitation direct, special, indirect, or consequential damages that may result from the use of these materials. SAP does not warrant the accuracy or completeness of the information, text, graphics, links or other items contained within these materials. SAP has no control over the information that you may access through the use of hot links contained in these materials and does not endorse your use of third party web pages nor provide any warranty whatsoever relating to third party web pages.

Any software coding and/or code lines / strings ("Code") included in this documentation are only examples and are not intended to be used in a productive system environment. The Code is only intended better explain and visualize the syntax and phrasing rules of certain coding. SAP does not warrant the correctness and completeness of the Code given herein, and SAP shall not be liable for errors or damages caused by the usage of the Code, except if such damages were caused by SAP intentionally or grossly negligent.

#### **Documentation in the SAP Service Marketplace**

You can find this documentation at the following address:

<http://service.sap.com/xi> -> **Media Library** -> **Documentation**

## Typographic Conventions

Type Style	Represents
Example Text	Words or characters that appear on the screen. These include field names, screen titles, pushbuttons as well as menu names, paths and options.
	Cross-references to other documentation
Example text	Emphasized words or phrases in body text, titles of graphics and tables
EXAMPLE TEXT	Names of elements in the system. These include report names, program names, transaction codes, table names, and individual key words of a programming language, when surrounded by body text, for example, SELECT and INCLUDE.
Example text	Screen output. This includes file and directory names and their paths, messages, names of variables and parameters, source code as well as names of installation, upgrade and database tools.
Example text	Exact user entry. These are words or characters that you enter in the system exactly as they appear in the documentation.
<Example text>	Variable user entry. Pointed brackets indicate that you replace these words and characters with appropriate entries.
EXAMPLE TEXT	Keys on the keyboard, for example, function keys (such as F2) or the ENTER key.

## Icons

Icon	Meaning
	Caution
	Example
	Note
	Recommendation
	Syntax

<b>1</b>	<b>Introduction</b>	<b>5</b>
1.1	Design Objectives	5
<b>2</b>	<b>Categorizing Items in the SLD</b>	<b>5</b>
2.1	Software Catalog - Products	5
2.2	Software Catalog - Software Component	8
2.3	Landscape – Technical Systems	10
2.4	Landscape – Business Systems	10
<b>3</b>	<b>Design Time: Objects in the Integration Repository</b>	<b>12</b>
3.1	Namespaces for Software Component Versions	12
3.2	Integration Scenario	13
3.3	Integration Processes	13
3.4	Interface Objects – Data Types	14
3.5	Interface Objects – Message Types	15
3.6	Interface Objects – Message Interface	15
3.7	Interface Objects – External Definition	16
3.8	Interface Objects – Data Type Enhancement	16
3.9	Interface Objects – Context Object	16
3.10	Mapping Objects – Message Mappings	16
3.11	Mapping Objects –Interface Mapping	17
3.12	Mapping Objects –Mapping Templates	17
3.13	Mapping Objects –Imported Archives	17
<b>4</b>	<b>Configuration Time: Integration Directory</b>	<b>18</b>
4.1	Configuration Scenarios	18
4.2	Party	18
4.3	Business Service	18
4.4	Integration Process Service	18
4.5	Communication Channel	19
4.6	Value Mappings	19

## 1 Introduction

This document provides naming conventions for objects in the System Landscape Directory (SLD), Integration Repository, and Integration Directory.

Consider the naming conventions provided as recommendations only. If you choose to deviate from the recommendations, ensure that you nevertheless always follow the same general principle described.

### 1.1 Design Objectives

Consider the following design objectives during the development process:

- Keep a list of all published functions and services.
- Publish generic, standard, reusable functions and services.
- Keep business systems linkage at runtime transparent.
- Organize your development packages and systems in such a way that development and transport are simplified.
- Ensure that previously developed items can be localized and identified easily.
- Make monitoring understandable and easily identifiable.
- Ensure a clear role separation exists between the interface owner and the mapping programmers.

To ensure that these guidelines are applied, you must ensure that you structure the objects correctly, use appropriate names, and provide organizational support.

This how-to guide provides naming conventions for objects in the Integration Repository, Integration Directory, and SLD.

Elements are proposed where applicable. These are marked by angular brackets (< >). Other elements are optional because they are considered to be applicable only in certain situations. These are marked by square brackets ([ ]). Where elements are optional, or where several proposals are given, analyze which solution is most appropriate for your installation, and then always use it in the same way.

We recommend that you create a set of example objects as a guide so as to simplify the procedure and to help developers remember the proposed naming conventions.

## 2 Categorizing Items in the SLD

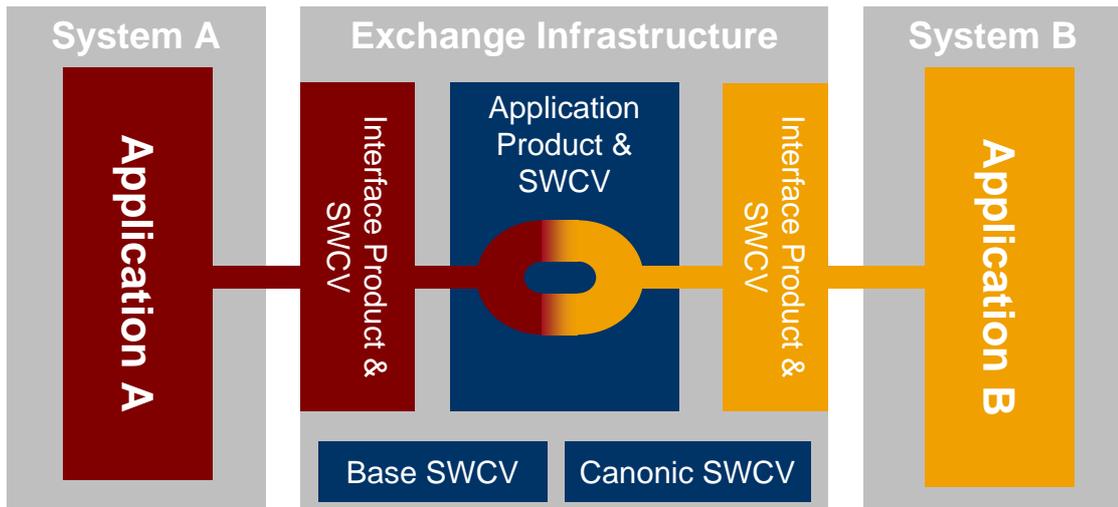
The SLD contains a software catalog with products, software components, and their respective versions.

### 2.1 Software Catalog - Products

We divide the products into five different groups (see figure below):

- **Interface Products:** Represent extensions to current applications. These extensions are imported later to the Integration Repository and connect the corresponding interface objects and integration scenario actions.

- **Exchange Infrastructure (XI) Application Products:** Represent objects in the Integration Repository other than interface objects (for example, mapping objects and integration processes).
- **Non-Interface Application Products:** Represent all other applications, including WebDynpro customer developments, SAP software, and so on.
- **Basis Products:** Ideal for templates, generic structures, shareable Java programs, and so on.
- **Canonical Definitions:** For generic business objects; intended for reuse.



To allow this differentiation, a prefix letter is added to the name.

Group of Products	Prefix
Interface Products	I
XI Application Products	A
Basis Products	B
Canonical Definitions	C

The product is on a very high level in the name hierarchy. Since naming conventions are derived (linked) from the product name, adding component-specific information will result in long strings. The following two guidelines will help you to create shorter names:

- Use abbreviations.
- Use codes for products. To do so, create a table of codes for products. This procedure is also recommended for integration scenarios and configuration scenarios.

Example:

Product Code	Description
PF024	ProdFactory – Production System – Stock Handling
CF0245	Finance2000 – Corporate Finance – Account Master Handling

Consider the following recommendations:

- In the case of interface products, it is the application software and not the technical environment that is the product.

For example, a customer develops a financial application in Java on a UNIX operating system and DB2 database. The product is the financial application developed by the customer, and not the operating system, programming language, or database.

- In the case of XI application products, the product is the most generic integration scenario that it will contain.

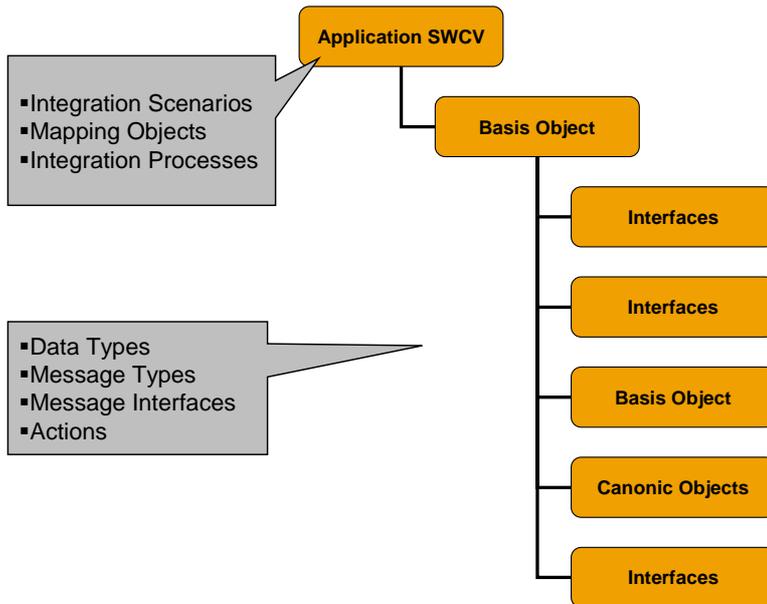
**Naming**

Third Party Products		
Vendor	Naming	<Company Domain Name>
	Example	MyComp.com
Name	Naming	[Group][Company][Vendor]<Product   Product Code>
	Example	I_MyCmpFin: Customer Financial System A_MyCmpARec: Account Receivables Integration A_MyCmpMM021: Material master in MDM C_MasterData: Canonical Object for Master Data Objects B_Common: Common routines I_MyCmpQM300: Quality Management Integration Product
	Details	<ul style="list-style-type: none"> <li>• For XI application products, the name of the product should represent the integration scenarios contained.</li> <li>• Use <i>Group</i> for the central developments of a holding.</li> <li>• Do not use <i>Company</i> if this is a global <i>Group</i> solution.</li> <li>• Use <i>Vendor</i> if this is not your own development.</li> </ul>
Version	Naming	Related Application Product
	Example	1.0.3

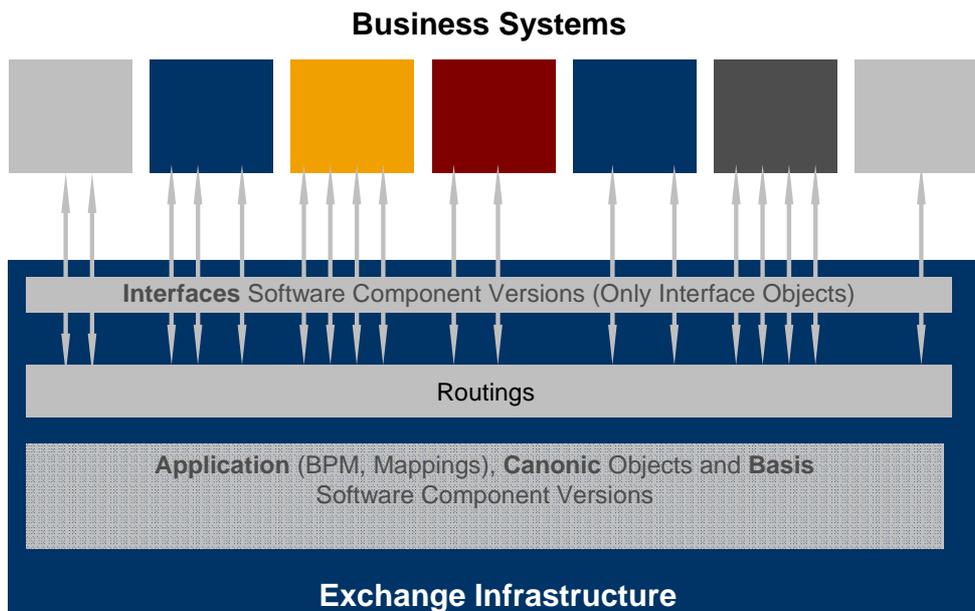
Third Party Products Based on SAP Products		
Vendor	Naming	Company Domain Name
	Example	MyComp.com
Name	Naming	I_SAP<Sap Product><Application Hierarchy>[Description]
	Example	I_SAPR3FiAp: Accounts Payable.
	Details	The group is always I because it represents an extension to SAP software.
Version	Naming	Related Application Product
	Example	1.0.3

## 2.2 Software Catalog - Software Component

The software component (SWC) and the software component version (SWCV) are basically categorized in the same way as the product (see 2.1). This means that software component versions will be grouped into interface software component versions, XI application software component versions, canonic objects software component versions, and basis software component versions (see figure below).



The SWC must also have a similar name to the product (see figure below).



Interface, basis, and canonic SWCs are to be included in the SLD as dependencies at installation time for the appropriate (XI) application SWC.

**Naming**

<b>Third Party Software Components</b>		
Vendor	ASAP Standard	N/A
	Naming	Company Domain Name
	Example	MyComp.com
Name	Naming	[GROUP_][COMPANY_][VENDOR_]<PRODUCT>_<APPLICATION>
	Example	I_MYCMP_FIN_ACC: Customer Financial System Accounting Interface  A_MYCMP_AREC_RP: Account Receivables, Accounting Replications  I_MYCMP_QM300_CHADM: Quality Management, Quality Characteristics Administration.  C_MD_MAT Canonical Object for Materials  B_ERROR_HANDLING: Common routines for Error Handling
	Details	Keep as short as possible.
Version	Naming	User-defined, based on the product scheme
	Example	1.0.3

<b>Third Party Software Components Based on SAP Products</b>		
Vendor	Naming	Company Domain Name
	Example	MyComp.com
	Details	N/A
Name	Naming	<PRODUCT>_<APPLICATION>
	Example	I_SAP_R3_FIAP_WS: Accounts Payable Web Services.
	Details	N/A
Version	Naming	User-defined, based on the product scheme
	Example	1.0.3
	Details	N/A

## 2.3 Landscape – Technical Systems

### Naming

SAP Technical Systems		
Name	Naming	SAP System Name
	Example	DES
	Details	N/A

Third Party Technical Systems		
Name	Naming	N/A
	Example	FINHOST;MAINFRM
	Details	Related to hostname but depends on the technology

## 2.4 Landscape – Business Systems

### Naming

SAP Business Systems		
Name	Naming	Sys_<Current SAP ALE Name>
	Example	Sys_DES_100
ALE Name	Naming	<Related SAP ALE Name>
	Example	DESCLNT100, DES_100

Third Party Business Systems		
Name	Naming	Sys_<Environment>[Company][Organizational Area]<MainProduct   AppRef>
	Example	Sys_DMyCompCommSales: Development environment for Sales System (commercial office) Sys_PMyCompBudgetMaster: Production environment for <i>Budget Master</i> financial application.
	Details	<ul style="list-style-type: none"> <li>• List of possible environments:                             <ul style="list-style-type: none"> <li>○ D: Development</li> <li>○ Q: Quality Assurance</li> <li>○ P: Production</li> <li>○ T: Training</li> </ul> </li> <li>• The organizational area stands for commercial, logistics, human resources, purchasing or financial areas for example.</li> </ul>
ALE Name	Naming	<Environment(1)><Company(3)><BusinessRole(3)><MainProduct(3) AppRef(3)>
	Example	DMyComSal

		PMycFinBgm
	Details	Maximum 10 characters, related to the business system name.

## 3 Design Time: Objects in the Integration Repository

### 3.1 Namespaces for Software Component Versions

There are two kinds of namespace:

- **Namespace based on business processes:** Namespace is global and independent from the organizational unit inside the company.
- **Namespace based on organizational units:** Namespace implements different solutions for the company's organizational units.

You can also mix both namespace types as required.

#### Naming

Namespaces Based on Business Processes		
Name	Naming	urn:<Company URN>:<Process Level N >:<Process Level N + 1>:<SWC>
	Example	urn:mycomp-com:finance:accrec:a_arec_rp: Accounts receivables in finance
	Details	<ul style="list-style-type: none"> <li>• Based on the product.</li> <li>• SWC: Remove the company. Example: SWC <i>A_MYCMP_AREC_RP</i>, becomes <i>A_AREC_RP</i>.</li> </ul>

Namespaces Based on Organizational Units		
Name	Naming	urn:<Company URN><SWC>:<OrgUnit Level N >:<OrgUni Level N + 1>
	Example	urn:mycomp-com:a_arec_rp:europa:finserv: Accounts receivables in for financial services in Europe
	Details	<ul style="list-style-type: none"> <li>• Based on the product.</li> <li>• SWC: Remove the prefix and company. Example: SWC <i>A_MYCMP_AREC_RP</i>, becomes <i>A_AREC_RP</i>.</li> </ul>

## 3.2 Integration Scenario

### Naming

Integration Scenario		
Name	Naming	[Code_]<BusinessProcessDescription>
	Example	DistributedAccountReceivable MM01_MaterialMasterMaintenance QM300_CharacApproval: Quality Management, quality characteristics approval.
	Details	<ul style="list-style-type: none"> <li>• Capitalize the first letters of each word.</li> <li>• Focus more on the process than the integration requirement. For example, show the process that requires replications instead of creating a 'replication scenario'.</li> <li>• It is also advisable to code the integration scenarios. If the application product (A_) is coded, it can be derived from the product.</li> </ul>
Action	Naming	<Business Object><Action   Event>[Description]
	Example	VendorCreated BillOfLadingModify
	Details	<ul style="list-style-type: none"> <li>• Events mostly refer to asynchronous interfaces.</li> <li>• Actions mostly refer to synchronous interfaces.</li> <li>• Must be created in the SWC that contains the related interface (not in the application SWC).</li> </ul>

## 3.3 Integration Processes

### Naming

Integration Processes		
Name	Naming	<Object><General Action> <Integration Scenario>
	Example	GoodIssuesDistribute AccountingPostingsReplicate MM01_MaterialMasterMaintenance
	Details	<ul style="list-style-type: none"> <li>• Scope: Prefix <b>p</b> for process global.</li> <li>• Use plurals to denote multiple line elements.</li> <li>• Use message type (business object type) if the container element is based on an interface.</li> </ul>
Container element	Naming	<Scope><Message Type>
	Example	pVendor Materials
	Details	<ul style="list-style-type: none"> <li>• Scope: Prefix <b>p</b> for process global.</li> <li>• Use plurals to denote multiple line elements.</li> <li>• Use message type (business object type) if the container element is based on an interface.</li> </ul>
Correlation	Naming	<Description>

	Example	CustomerCode TaxIdentifier
	Details	Use business names, not technical names.
Step name	Naming	<Object Variable><Step Type   Actual Operation>
	Example	pInvoicesSplit pCustomerReceive DangerousMaterialsSend
	Details	<i>Actual Operation</i> represents a more specific explanation.

### 3.4 Interface Objects – Data Types

The following naming conventions apply to data types which do not reference existing external data formats from external systems. For example, if you design the interaction with a database, it is recommended that you name your data type in accordance with the name of the table that provides or receives the information instead of creating a new name.

Inbound SAP RFC and IDoc names (imported objects) must not be renamed or copied to new interface objects.

Note that in general, data types refer to object types (for example, orders, customers, and accounting documents) rather than object methods or events (for example, create order, update customer, and change date). Methods or events in general are represented by message interfaces. Consequently, the data type name should represent an object type. Furthermore, a data type should represent an 'application object' rather than a technical object (for example, SalesOrder instead of SalesOrderTable).

It is also recommended that you keep the data type name short since it will be used in other object names.

#### Naming

Data Types		
Name	Naming	[prefix]<Business Object>[suffix]
	Example	Customers, cCustomer, SelectCustomer, SelectCustomersResp
	Details	<ul style="list-style-type: none"> <li>• Prefixes:                             <ul style="list-style-type: none"> <li>○ c for canonic formats</li> <li>○ Adapter-oriented data types with specific usage and formatting rules:                                     <ul style="list-style-type: none"> <li>▪ Select: JDBC Select</li> <li>▪ Insert: JDBC Insert</li> </ul> </li> </ul> </li> <li>• Suffixes:                             <ul style="list-style-type: none"> <li>○ Adapter-oriented data types with specific usage:                                     <ul style="list-style-type: none"> <li>▪ Resp: Response</li> </ul> </li> </ul> </li> <li>• Use plurals for multiple instance data types.</li> </ul>

## 3.5 Interface Objects – Message Types

### Naming

Message Types		
Name	Naming	<Data Type>
	Example	Customer, cCustomer, SelectCustomer, SelectCustomerResp, CustomerResp

## 3.6 Interface Objects – Message Interface

The message interface (except for abstract interfaces) is related to an 'action' applied to an object (or a method), or an event triggered or received.

### Naming

Message Interfaces		
Name	Naming	<Category><Mode>_<Business Object>[<Action   Event>]
	Example	os_CustomerSend oa_ClosingPeriodProcessingFinished ia_MaterialActivate aa_Customer
	Details	<ul style="list-style-type: none"> <li>• <i>Action</i> or <i>Event</i> does not apply to abstract interfaces.</li> <li>• Example actions: List, Detail, Save, Send, Create, Modify, Delete, Activate (verbs in infinitive).</li> <li>• Example events: Finalized, Saved, Started (Verbs in participle).</li> <li>• Categories: <ul style="list-style-type: none"> <li>○ o: Outbound</li> <li>○ i: Inbound</li> <li>○ a: Abstract</li> </ul> </li> <li>• Mode <ul style="list-style-type: none"> <li>○ s: Synchronous</li> <li>○ a: Asynchronous</li> </ul> </li> <li>• The purpose of using both Category and Mode as prefixes is to keep categories grouped in the Integration Repository. Change the sequence of the components, for example use both concepts as suffixes, if you prefer to keep different interfaces for the same object together in the Integration Repository.</li> </ul>

## 3.7 Interface Objects – External Definition

### Naming

No specific naming rules exist. Use descriptive names and capitalize the first letters of words.

## 3.8 Interface Objects – Data Type Enhancement

### Naming

Data Type Enhancements		
	Naming	<Basis object name>_<Description>
Name	Example	Customer_Extensions Material_WarehouseData

## 3.9 Interface Objects – Context Object

### Naming

Context Objects		
	Naming	<Description>
Name	Example	CustomerCode TaxIdentifier
	Details	Use business names, no technical names.

## 3.10 Mapping Objects – Message Mappings

Message mappings act as conversion programs between source and target formats. This means that the most natural way to name them is to use the source and target messages. The problem is that the name then exceeds the maximum length because to identify the message type properly, you must use the SWC as the prefix for both the source and target message.

### Naming

Message Mappings		
	Naming	[bpm_]<SourceMsgType#1>[_and_]<SourceMsgType#N>[_to_]<TargetMsgType>.
Name	Example	Customer_to_SelectCustomer SelectCustomerResp_to_Customers bpm_InvoiceHeaders_and_InvoicePositions_to_Invoices Materials_to_MATMAS_MATMAS03
	Details	<ul style="list-style-type: none"> <li>Do not include the software component in the name. Involved software components are easily determined by the basis object list.</li> <li>For SAP IDocs replace the dot by an underscore.</li> </ul>

### 3.11 Mapping Objects –Interface Mapping

#### Naming

Interface Mappings		
Name	Naming	<Most representative message mapping name>
	Example	Customer_to_SelectCustomer SelectCustomerResp_to_Customers Materials_to_MATMAS_MATMAS03
	Details	For synchronous interfaces, use the request (not the response) message mapping name.

### 3.12 Mapping Objects –Mapping Templates

#### Naming

Mapping Templates		
Name	Naming	tmpl_<Message Mapping name>
	Example	tmpl_CustomerHeader_to_SelectCustomer tmpl_Materials_to_MATMAS_MATMAS03

### 3.13 Mapping Objects –Imported Archives

#### Naming

Imported Objects		
Name	Naming	<Message Mapping name>_<TransformationTechnology>
	Example	Customer_to_SelectCustomer_Java SelectCustomerResp_to_Customers_XSLT
	Details	Transformation Technologies: <ul style="list-style-type: none"> <li>• Java</li> <li>• XSLT</li> </ul>

## 4 Configuration Time: Integration Directory

### 4.1 Configuration Scenarios

#### Naming

Configuration Scenarios		
Name	Naming	<Integration Scenario>
	Example	DistributedAccountReceivable MM01_MaterialMasterMaintenance QM300_CharacApproval: Quality Management, quality characteristics approval.

### 4.2 Party

#### Naming

Parties		
Name	Naming	<Description>
	Example	MyPartner SupplierX

### 4.3 Business Service

#### Naming

Business Services		
Name	Naming	Srv_<Related Organizational Area   Customer ID   System ID>
	Example	Srv_Intranet: General intranet services Srv_MyCustomer : Services for MyCustomer (without using party) Srv_Transportation : Web services for transportation

### 4.4 Integration Process Service

#### Naming

Business Processes		
Name	Naming	Prc_<Integration Process>
	Example	Prc_GoodIssuesDistribute Prc_AccountingPostingsReplicate Prc_MM01_MaterialMasterMaintenance

## 4.5 Communication Channel

Naming the channels properly requires knowledge of the channel type.

Another necessary consideration is that the name of the channels can be generic or detailed based on the technology. For example, one server RFC channel from a SAP system is enough to process any type of requests, but since the JDBC adapter uses the SQL string, you must create more than one channel to work with the same database. Therefore, in many situations it makes sense to add the message types to the name.

It often makes sense to consider either the direction in the information flow or the requester. For example, system A contacts system B twice, first to replicate a purchase order and later to request a customer description. Both times the requester is the same, but the main flow conceptually changes. In this case it is recommended to use the 'requester' point of view (since it is more constant).

### Naming

Communication Channels			
Name	Naming	Sender	<Service>_<Message Type>_<Adapter Type>_Server
		Receiver	[Ack_]<Service>_<Message Type>_<Adapter Type>_Client
	Example		Sys_PRD_810_IDoc_Client Sys_XR3_100_RFC_Client Srv_Logistics_HTTP_Client: Posts http requests to the Logistics Business Service. Sys_PMyCompHRPayMaster_Payroll_JDBC_Server: Polls the production HR database to recover the payroll records.
	Details		<ul style="list-style-type: none"> <li>• Senders are servers that poll or listen in a port.</li> <li>• Clients log on to recipients of requests.</li> <li>• By including the service, monitoring in the adapter monitoring in the RWB will be improved.</li> <li>• Use Ack prefix for IDoc acknowledgements.</li> </ul>

## 4.6 Value Mappings

### Naming

Value Mappings		
Group	Naming	<Business Object Descr.> - <BO Instance Descr.>
	Example	Business Unit – Cars Currency – Dollar Currency – Euro Country – Switzerland
Agency	Naming	<Agency Name   Generic Agency Name>
	Example	DES_100 Sys_PMyCompBudgetMaster
Scheme Namespace	Naming	urn:<company urn>:[Organizational Unit]:<generic agency id>
	Example	urn:mycmp-com:fin:r3

		urn:mycmp-com:prd:prdsys urn:custcmp-com:fin:budgetmaster
	Details	In the case where the agency is a business system, use a generic namespace determined independently from the environment (omits the prefixes or uses the generic product).
Scheme Object Type	Naming	<XML field element name>
	Example	BUS_AREA Division TaxCode MANDT