1. **Naming Conventions and Guidelines**

   Standards are mandatory practices which are to be followed in the implementation of SAP objects.Naming conventions are subjective and can be applied in multiple ways. The defined conventions are a good starting point and, if necessary, can be extended or modified.

   Additionally, all Cloud Integration objects should be named using **Apache Camel Case**, separating words by uppercase letters if not mentioned otherwise in this document.

Advanced Event Mesh

Object Type	Definition	Proposed Naming Convention	Example

Topic		Lower Case or camelCase
		`<`**`Sector`**`>/<`**`Source`**`>/<`**`BusinessObject`**`>`
		`<`**`Sector`**`>/<`**`Source`**`>/<`**`BusinessObject`**`>/<`**`Verb`**`>/<`**`Version`**`>/<`**`Properties`**`>`
			`im/time/material`
			`mt/smart/salesorder`
			`im/time/material/created/v1/HAR17F`

Queue		**`q`**`/<`**`Sector`**`>/<`**`Source`**`>/<`**`BusinessObject`**`>/<`**`Target`**`>`	`q/IM/TIME/Material/JJCC`
Dead Message Queue		**`dmq`**`/<`**`Sector`**`>/<`**`Source`**`>/<`**`BusinessObject`**`>/<`**`Target`**`>`	`dmq/IM/TIME/Material/JJCC`

Integration Suite

Object Type	Definition	Proposed Naming Convention	Example
Custom Integration Package	A collection of artifacts with integration content (integration flows, value mappings, APIs, and so forth) belonging to business area, packaged, and delivered together.	**Name**: The name of the package should refer to the two products plus product lines between which the integration needs to take place. If you are developing generic integration packages or country specific packages, then refer to generic and country specific sections in the example. Package Naming - We will use - **Sector_<ProcessArea>_<Source>_<Target>_<BusinessObject>** Note, the ordering of Source and Target in package name should not imply directionality or request flow. Rather, it should simply be consistent within your project. iFlow names are where directionality (e.g., Source to Target, or Target to Source) should be used. **List of applicable Sector** • IM • MedtechTranscend or MT • MedtechNonTranscend or MNT **List of applicable ProcessArea** • Source • Plan • Make • Deliver • Quality • Finance	**`IM_Deliver_OMP_CFIN`** **`MT_Source_JJCC_S4Hana`**
		Common: Common (aka generic) package for Value Mappings, Common reusable IFlows **`<Sector/ProcessArea/Project>_Common_<ContentDescription>`**	**`Invoices_Common_S4HanaApis`**

		Version: The version number should always be 1.0.0 when the package is productionized and should be incremented by +1.0.0 to either 1.1.0 or 1.0.1 depending on Transport change is major or minor	Version: 1.0.0(when productionized) ,1.0.1 (After micro change of first transport of artifact) on whether change is major or minor or micro.
		Overview: The full long description of the package describing the usage, functionality, and goal of the package.	Overview:This package enables the integration of business processes between SAP S/4HANA or SAP S/4HANA Cloud with SAP SuccessFactors
		TAGS: The package should be tagged with country and line of business and industry using relevant dropdowns and a common keyword search should be based onmost usedin package short description or Interface ID(S) of package.	Country:Canada Line of Business: Sandbox Industry: JnJ Keyword: Customer Product: SAPS/4 HANA Project Name: “JnJ Sandbox"
iFlow **--** **Integration Flow**	BPMN based model that is executable by orchestration middleware. An iflow/ Integration flow allows you to specify how a message is processed on a tenant	**IF_<BusinessObject>_<InterfaceAction>_<Sender>_<Receiver>** **IF_<RICEFW_ID>_<BusinessObject>_<InterfaceAction>_<Sender>_<Receiver>** **IF_<BusinessObject>_<InterfaceAction>_<Sender>** **IF_<StandardObjectType>_<InterfaceAction>_<Sender>** **IF_Common_<InterfaceDescription>** **BusinessObject Examples** ProcessOrder, MaterialMaster, SalesOrder, BusinessPartner **StandardObjectType Examples** MasterData, SalesOrderTypes, CSVData **InterfaceAction Examples** Send, Receive, Create, Publish, Route, Dispatch, Replicate, Insert, Maintain or POST, GET, DELETE (CRUD Operations).	**iFlow Example Names** **IF_CreateSalesOrder_POST_S4HANA** **IF_Customer_Send_TIME_SFDC** **IF_MasterData_Dispatch_S4HANA_AEM** **IF_GetConnectionID_ELIMS**
Sender	Sender system Endpoint **Note, this is not the same as the SAP_Sender message header**	**S_<IntegrationType>_<SenderSystem>** Integration Types: OP: On Premise System CP: Cloud Providers B2B: Business Partners	**S_CP_SFSF** **S_S4HANA**
Receiver	Receiver System Endpoint **Note, this is not the same as the SAP_Receiver message header**	**R_<IntegrationType>_<ReceiverSystem>** Integration Types: OP: On Premise System CP: Cloud Providers B2B: Business Partners	**R_OP_ERP** **R_S4HANA**
Integration Process	Main Process flow / container to include all Integration steps	**IPR_<Business Process description that the process aims to achieve>**	**IPR_PurchaseOrderCreation** **IPR_MasterDataRepliction**
Local Integration Process	Sub Process flow / smaller fragments to simplify integration process	**LIP_<Local business process flow that the process aims to achieve>**	**LIP_UpdateMasterData** **LIP_GetSalesOrder**
Exception Subprocess	Use Exception sub-process to catch any exceptions thrown in the integration process & handle them	**EXC_SP_<Exception process description>**	**EXC_SP_Email_Notification**

ID Mapping	ID Mapping: generated are Target Message ID that is uniquely associated with certain parameters of the inbound message. Target Message ID is Global unique ID that identifies target message	**ID_<SourceMessage>[<Format>] _to_<Target Message>[Format]**	**ID_PurchaseOrder_Split** **ID_POGUID_XML_to_POGUID_JSON**
Message Mapping	Message Mapping: Artifact to map equivalent fields in the Source & Target systems	**Full Version** **MM_<SourceMessage>[<Format>] _to_<TargetMessage>[Format]** **Shorthand Version** (use only when the business object being mapped remains fundamentally the same) **MM_<BusinessObject>_<Source>[<Format>] _to_<Target>[<Format>]** **Shorthand for Validate Payload Framework Message Map** **MM_<BusinessObject>_<Source /Sender|Format>_ValidatePayload**	**MM_PurchaseOrderS4HanaXML_to_PurchaseOrderC4CJson** **MM_PurchaseOrderS4Hana_to_PurchaseOrderC4C** *Shorthand example:* **MM_InspectionLot_TIMEXml_to_ELIMSJson** *Shorthand for Validate Payload Framework Message Map:* **MM_SalesOrderCreate_JJCC_ValidatePayload**
Operation Mapping	Operation Mapping: Relates an outbound service interface operation with an inbound service interface operation	OM_<SourceMessage>[<Format>] _to_<TargetMessage>[Format]	**OM_PurchaseOrderIDOC_to_PurchaseOrderRFC**
XSLT Mapping	XSLT Mapping: XSLT implements Xpath expressions to select sub-structures of an XML Document. Using templates in XSLT, one can define the mapping rules for the selected substructure	XSLT_<SourceMessage>[<Format>] _<Function>	**XSLT_PurchaseOrderXML_Split**
Value Mapping	Value Mapping: Used to map source system values to target system values.	VM_<SourceAgency>_to_<TargetAgency>_<FieldName>	**VM_Ariba_to_S4Hana_OrderStatus**
Content Modifier	Extract or Modify message content	CM_<ModificationDetails>	**CM_LogHttpHeader** **CM_GetPayloadDetails**
Converter	Converters allows message conversion from one format to another format	CONV_<SourceFormat>_to_<TargetFormat>_<MessageAdditionalInfo>	**CONV_CSV_to_XML_OrderDetails** **CONV_EDI_to_XML_EmployeeData** **CONV_JSON_to_XML_MaterialDetails**
Encoder	Encode messages using an encoding scheme to secure any sensitive message content during transfer over the network. Can be used for compressing messages as well	ENC_<EncodingType>_<EncodedMessageDetails>	**ENC_Base64_UserCreditDetails** **ENC_MIME_InvoiceDetails** **ENC_ZIP_CompressEmployeeeDetails**
Decoder	Decode the encoded message content using a decoding scheme. Can we used for decompressing the compressed message as well	DEC_<EncodingType>_<DecodedMessageDetails>	**DEC_Base64_UserCreditDetails** **DEC_MIME_InvoiceDetails** **DEC_ZIP_DecompressEmployeeeDetails**
Filter	Condition based removal or filtering of elements from a given message node	FLT_<FilterDataInformation>	**FLT_PurchaseOrderNumber** **FLT_PayloadData**
Script	Custom Scripts can be written for complex transformations that can't be achieved using existing options provided in the SAP IS palette. Scripting in SAP Integration Suite can be done using Groovy Script & Java Script	SCR_<ObjectiveOfScript>	**SCR_LogPayloadResults** **SCR_FormatMessage**
Request Reply	Call an external system / external service and get back a response	RR_<RequestedSystem>_<RequestedInformationDetails>	**RR_Hybris_GetAccountPayableDetails** **RR_TMS_GetPaymentAcknowledgement**
Send	Send step can be used to configure a service call to an external Receiver system where no reply is expected	SND_<InformationToBeSent>_to_<TargetSystem>	**SND_UserBankDetails_to_S4Hana** **SND_CreditCardData_to_S4Hana**

Aggregator	Combine the incoming chunks of messages into one single message. The Aggregator will do multi-mapping while combining the messages. The messages can be combined with/without any order	AGGR_<MessageDetails>	**AGGR_ItemsWithSameOrderNumber**
Gather	Gather step merges from different routes into a single message with the option to define certain strategies on how to combine the initial messages	GAT_<MessageDetails>	**GAT_SalesOrderDetails**
Join	Join elements enables you to bring together messages from different routes before combining them into a single message. Join is used in combination with Gather. In case if the message was split using Splitter, then we don't need a join. In the latter case, Gather can be used without join	JOIN_<MessageDetails>	**JOIN_SalesOrderDetails**
Multicast	Send copies of the same message to multiple routes. The copies can be sent all at once for parallel processing using Parallel Multicast or in a sequence using Sequential Multicast	Sequential Multicast: SCAST_<MessageDetails> Parallel Multicast: PCAST_<MessageDetails>	**SCAST_SalesOrderDetails** **PCAST_SalesOrderDetails**
Router	Routes the message into multiple paths based on condition	RTR_<RoutePropertyDetails>	**RTR_SalesOrderType** **RTR_EmployeeStatus**
Splitter	Allows to break the message into smaller parts, that can be processed independently	SPLIT_<SplitPropertyDetails>	**SPLIT_OrderItemData**
Encryptor	Allows to convert plain text into ciphered text	ENC_<EncryptionType>_<EncryptedMessageDetails>	**ENC_PGP_BankDetails** **ENC_PKCS7_USerCreditCardData**
Decryptor	Allows to convert ciphered text back to its original plain text	DEC_<DecryptionType>_<DecryptedMessageDetails>	**DEC_PGP_BankDetails** **DEC_PKCS7_UserCreditCardData**
Signer	Signer allows the participants to know their identity & ensure the authenticity of the messages being sent on the cloud. It guarantees identity by signing the messages with one or more private keys using a signature algorithm.	SIG_<SignerType>_<SignedMessageDetails>	**SIG_PKCS7_BankDetails** **SIG_XMLDigital_UserCreditCardData**
Verifier	Verifier ensures the signed message received over the cloud is authentic	VERI_<VerifyType>_<VerifiedMessageDetails>	**VERI_PKCS7_BankDetails** **VERI_XMLDigital_UserCreditCardData**
Validator	The EDI Validator validates the message payload in EDI flat file format against the configured XSD schema The XML Validator validates the message payload in XML format against the configured XML schema	VAL_<ValidatorType>_<ValidationMessageDetails>	**VAL_EDI_SalesOrderData** **VAL_XML_EmployeeOrgData**
Message Header	Parameter that allows to define ways of controlling message processing. This is transferred as a part of message header	<CustomHeaderNameinCamelCase> <StandardHeader>	**messageID** **Content-Type** **X-CSRF-Token**
Exchange Properties	Parameter that allows to define ways of controlling message processing. Exchange properties is used to store additional data besides the message that is to be processed.	<CustomPropertyNameInCamelCase>	**elementCounter** **payloadLink**
End Point for Generic Iflows	Endpoints are the URL to trigger a specific integration flow. It must be unique to iflow in a tenant	<Country>/<BusinessProcess>/<InterfaceDescription>	**/Global/OTC/SalesOrderCreation** **/Global/MM/PurchaseOrderReplication**
Communication Channel	An integration flow channel allows you to specify which technical protocols are to be used to connect a sender or a receiver to the tenant	<Protocol>_<Direction>_<OperationDefinition>	**HTTPS_SND_PurchaseOrderCreate** **ODATA_SND_SalesOrderRead**
Security Material / Artifacts	Security artifacts are the credentials needed to connect to an external system	<Business Unit>_<SystemIdentifier>_<AdapterType>_<TypeOfSecurityMaterial> Global_<SystemIdentifier>_<AdapterType>_<TypeOfSecurityMaterial> Business Partner System: <System>_<Partner>_<AdapterType>_<TypeOfSecurityMaterial>	**S4H_QASCLNT100_Odata_SecureParameter** **SystemA_SFTP_UserCredentials** **FBS_SFTP_UserCredentials** **CIBC_SFTP_UserCredentials** **Global_QASCLNT100_OData_Oauth**

Key Stores	Consist of Certificates and key pairs that are stored in one keystore per tenant, referred to also as tenant keystore. A keystore is used to secure message exchange both at transport level and at message level	<SystemIdentifier>_<KeyStoreType>_<AddtionalDetails>	**SystemA_Certificate** **SystemA_SSH_KeyPair** **SystemA_RSA_KeyPair**
JDBC Material	JDBC Data Sources allow persistence in an external data base. The JDBC Data Sources allows you to create and manage a cluster of artifact connections to interact with a database (DB). Each data source contains information on database type, and database-specific configuration parameters.	<TypeOfDB>_<DataBaseID>	**HANA_DataBase1** **MSMySQL_DataBase1** **ASE_DataBase2**
User Roles	Custom roles can be used in runtime during inbound authorization of an integration flow (User Roles) or during monitoring to protect the business data of a subset of artifacts (Access Policies used)	Runtime Roles:ESBMessaging.send_<Expression/InterfaceDescription/QueueName>_RTCustomRole Monitoring Roles:<Expression/InterfaceDescription>_AccessPolicy_MONCustomRole **Standard Roles will be used for Iflows. Custom Roles wherever deemed appropriate to any use case will be created and leveraged.	**ESBMessaging.send_EmployeePayrollRead_RTCustomRole** **ESBMessaging.send_HRData_RTCustomRole** **EmployeePayrollRead_AccessPolicy_MONCustomRole** **HRData_AccessPolicy_MONCustomRole**
Message Queues	Messaging Queues enable persistence and asynchronous messaging in integration flows	**<AdapterType>_<IntegrationFlowID>**	**JMS_Update_SalesOrder_LS_to_S4Hana** **JMS_Send_PurchaseOrder_Ariba_to_OpenText**

2.1 Externalize Configuration Parameters

Externalize volatile configurations in an integration flow. Certain parameters of a processing step in an integration flow must be changed depending on the environment in which the integration flow is executed. For example, transporting an integration flow from the test tenant to the productive tenant usually requires reconfiguring the connectivity to the application system to which a message is sent. We refer to such parameters as volatile because they are to be adapted each time an integration flow is deployed in another landscape.

2.2 APIM Naming Conventions

Component Name	Proposed Naming Convention	Example
API Proxy Name (shown in API Portal)	API_<BusinessObject>_<Message Type Name>_<Action>_<Protocol Type>_v1	API_SalesOrder_Simulate_v1
API Proxy Base Path	/<Version Number>/<Business Object>	/v1/ProductionOrder
API Title (shows in Dev Portal)	Business Object Description	Production Order Confirmation and Cancellation
API Proxy Resource	Message Type/Process Name	POCancellation POConfirmation
Route Rule Name	Message Type/Process Name	POType
Target End Points	Message Type/Process Name	/POCancellation /POConfirmation
Product Name	System name/Business Process(S4)	MES_S2D
Application Name	Consumer System Name	MES
API Provider Name	<SystemID>_<EnvironmentName>/<System Name>	SDB_200, GBD_200, Salesforce
Key Value Map	<ProductName>_KVM	BTP_KVM, SBD_KVM, GBD_KVM

API & Artifact Naming Conventions

Basic Authentication	BA-Purpose	BA-TargetCredentials, BA-Decode
GenerateJWT	GJWT-Purpose	GJWT-GenerateAccessToken
Verify API Key	VAK-Purpose	VAK-VerifyAPIKey
Access Control	AC-Purpose	AC-AllowIPList
Invalidate Cache	IC-Purpose	IC-APIPermissions
Lookup Cache	LC-Purpose	LC-APIPermissions
Populate Cache	PC-Purpose	PC-APIPermissions
Spike Arrest	SA-Purpose	SA-Throttling
Raise Fault	RF-Purpose	RF-RatelimitViolation
Service Callout	SC-Purpose	SC-GetAPIPermissions
Oauthv2	OA-Purpose	OA-verifyAccessToken, OA-GenerateOAuthToken
Java Resource name	Purpose	verifyRoles.js (Use camel Case)
XSLT resource name	Purpose	replaceSessionId.js (Use camel Case)
WSDL resource name	Purpose	Order.wsdl
Message Validation	MV-Purpose	MV-ValidateSOAP
Open Connectors	OC-SystemName	OC-AzureBlob
Python Script	PS-Purpose	PS-SetVariables

Generic Integration Flow Guidelines

The following guidelines should be used to design integration flow layout for simplifying maintenance:

- Try to avoid overlapping sequence flows
- Avoid kinks (or confusing twist/turns)in the sequence and message flow connectors – try to keep them as straight as possible
- Avoid overlapping process steps – in case you need many process steps, tryexpanding the canvas and arrange the process steps neatly.
- Modularize wherever possible – in case of complex logics, try to break it down into small, easy to understand modules. Move the logic of each module into a sub-process. Name the sub-process appropriately to describe the module’s operation.
- Do not mix multiple transformations in a single script or sub-process – one sub-process should only contain the logic for one function.
- Do not assign the whole XML message to a header or a property unless necessary. Clear it once done.
- Always keep the flow direction from left to right. The sender always comes on the left and the receiver on right.

It is recommended to limit the total number of steps in integration flow to 10 and use the steps local integration process to modularize complex integration flows for better readability and ease of maintenance.