

Open Catalog Interface (OCI)



Release 5.0



© 2003, 2013 SAP AG or an SAP affiliate company. 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 propriety software components of other software vendors.

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.

SAP 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 other countries. Please see <http://www.sap.com/corporate-en/legal/copyright/index.epx#trademark> for additional trademark information and notes.

Icons in Body Text

Icon	Meaning
	Caution
	Example
	Note
	Recommendation
	Syntax

Additional icons are used in SAP Library documentation to help you identify different types of information at a glance. For more information, see *Help on Help → General Information Classes and Information Classes for Business Information Warehouse* on the first page of the any version of *SAP Library*.

Typographic Conventions

Type Style	Description
<i>Example text</i>	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	Technical names of system objects. These include report names, program names, transaction codes, table names, and key concepts of a programming language when they are surrounded by body text, for example, SELECT and INCLUDE.
Example text	Output on the screen. This includes 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	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. 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, F2or ENTER.

Table of Contents

1	Introduction.....	6
2	Calling Up the Catalog Using the SRM Server	7
2.1	Settings in the SRM Server	8
2.2	URL of the Product Catalog	8
2.2.1	Parameters.....	8
2.2.2	Return URL	8
2.3	Additional Functions in the Product Catalog.....	9
2.3.1	Detailed Display of a Product or Service.....	9
2.3.2	Validation of a Product.....	9
2.3.3	Sourcing/Product Search.....	10
2.3.4	Background Search.....	10
2.4	Overview of the Call-Up Parameters	10
3	Return from Catalog	11
3.1	Use of the Call-Up Parameters from Section 2.4	12
3.2	Fields and Field Checks	13
3.2.1	Required and Optional Fields.....	14
3.2.2	Product Numbers.....	14
3.2.3	Configurable Products.....	15
3.2.4	External Product Categories.....	15
3.2.5	Introduction of New fields to enable transfer of Item Hierarchy to SRM.....	15
3.2.6	Introduction of Inbound Handler and Outbound Handler	16
3.2.7	Introduction of Portal Navigation while transferring data to SRM.....	17
3.2.8	Customer-Specific Extensions in the OCI	17
3.3	XML Variant of the OCI.....	17
4	Handling of the Browser Window	18
5	Secure OCI.....	18
5.1	Launching Catalog with Secure OCI	18
5.1.1	Initial Handshake	19
5.1.2	Response to backend call.....	19
5.2	Return from catalog.....	20
5.2.1	Initiate backend call.....	20
5.3	Enablement of Secure OCI for catalogs.....	21
6	Enhancements in OCI 5.0 for Mass Upload	21

6.1	Mass replication of data delivered in a physical file	22
6.2	Mass replication of data over HTTP(s) connection	22
6.3	Replicating Attachments.....	22
6.4	Format for Data Transfer.....	23
6.5	Settings in the SRM Server.....	23
6.5.1	Define External Web Services.....	23
6.5.2	Activate SICF service for delivering images/attachments.....	23
6.5.3	Proxy server configuration in the BAdI SRM_GET_PROXY_INFO	24
6.5.4	Customer Enhancements	24
6.5.5	Additional settings for Mass Upload over Http(s).....	24
6.6	Function for mass upload over Http(s)	25
6.6.1	Technical Specification for JSON download	25
6.7	Additional Functions.....	32
6.7.1	Catalog Managed Items.....	32
6.7.2	Technical Specification for real-time Quantity check.....	33
6.8	Data transfer format and rules	33
6.8.2	Product Numbers.....	49
6.8.3	Configurable Products.....	49
6.8.4	External Product Categories.....	49
6.8.5	Customer-Specific Extensions in the OCI	50
7	Troubleshooting	50

1 Introduction

The Open Catalog Interface (OCI) incorporates external product catalogs into SRM Server applications. This way, data that is required to create shopping cart items in the SRM Server can be transferred directly from the external catalog to the SRM Server application. The mass upload feature is available with the release of Open Catalog Interface version 5.0, where it is possible to replicate the data from external catalogs into the SRM system using JSON (Java Script Object Notation) format over Hyper Text Transfer Protocol (HTTP) or physical file upload. This capability is offered only with the SAP SRM User Interface Add-On 1.0 for SRM Server. The OCI 5.0 standard adds the following capabilities on top of OCI 4.0:

- Catalog data replication to SRM so that the end users can search for items in all catalogs using SAP Enterprise Search without leaving the SAP SRM screen. Catalog service providers can choose which items are replicated to SAP SRM and to what extent. Details are provided in [section 6](#).
- Introduction of new fields to enable transfer of item hierarchy to SAP SRM. Details are provided in [section 3.2.5](#).
- Introduction of the inbound and the outbound handler. Details are provided in [section 3.2.6](#).
- Secure transfer of user credentials (Secure OCI). Details are provided in [section 5](#).

For a catalog to be OCI 5.0 compliant, it must also be OCI 4.0 compliant and support the features listed above.

SRM Server continues to adhere to the OCI version 4.0 and below, which offers the transfer mechanisms over Hyper Text Transfer Protocol (HTTP).

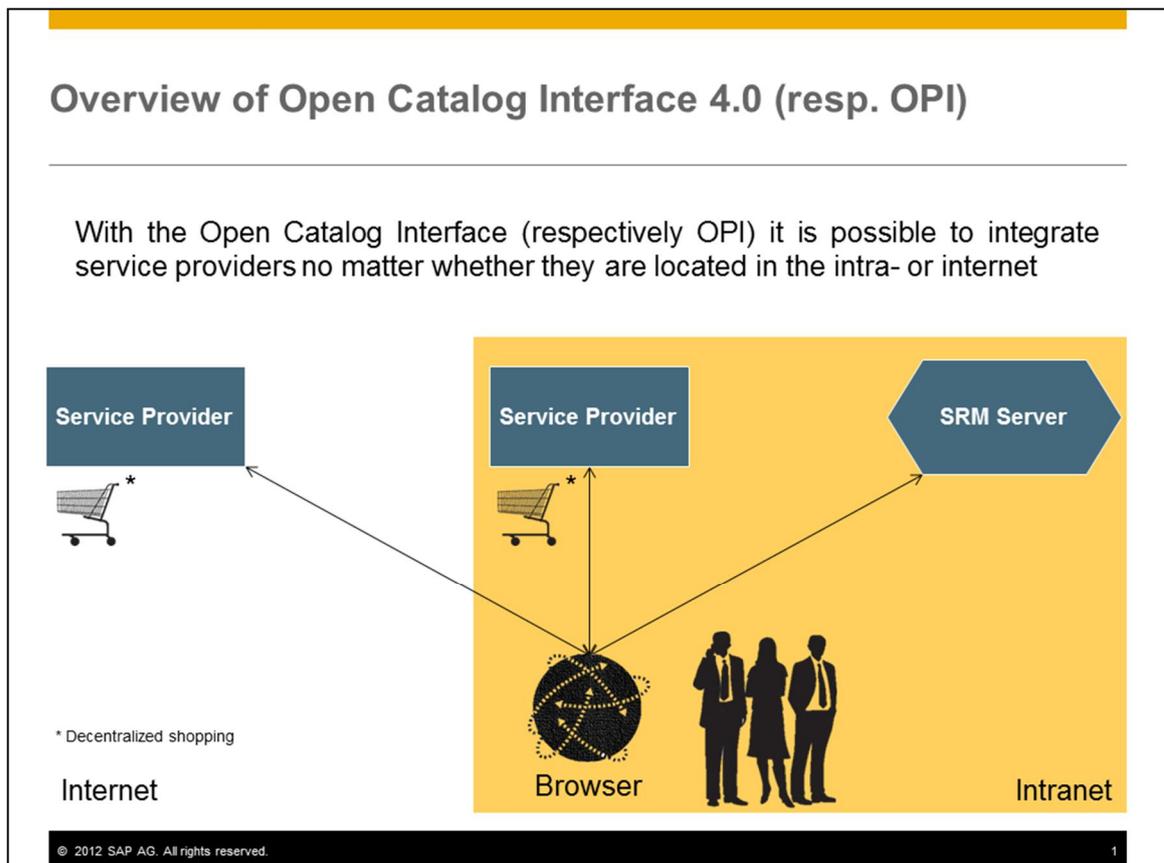


Figure 1 Overview of Open Catalog Interface 4.0

In this case, the user is working with an *SRM Server* application, which displays the available catalogs. The user calls up one of these, selects the required products, and then transfers the product data back to the SRM Server application.

This documentation describes both the architecture and the structure of the OCI and thus

provides all the information that is necessary to support the OCI with a product catalog. In addition, it describes the possible processes and their prerequisites. The documentation has been written both for producers of catalogs and for system administrators of *SRM Server* systems.

[Section 2](#) describes how the SRM Server calls the catalog as specified in OCI 4.0.

[Section 3](#) describes how the data is transferred from the catalog to the *SRM Server* as specified in OCI 4.0.

[Section 4](#) describes the handling of the different browser windows specified in OCI 4.0.

[Section 5](#) describes the additions made to Secure OCI.

[Section 6](#) describes enhancements that have been made for OCI 5.0.

2 Calling Up the Catalog Using the SRM Server

In order for a product catalog to be called up using the intranet or internet, its URL must be known in the SRM Server. If the product catalog requires additional parameters for the call-up (for example, logon names or a language identifier), these parameters also have to be known in the SRM Server before the call-up.

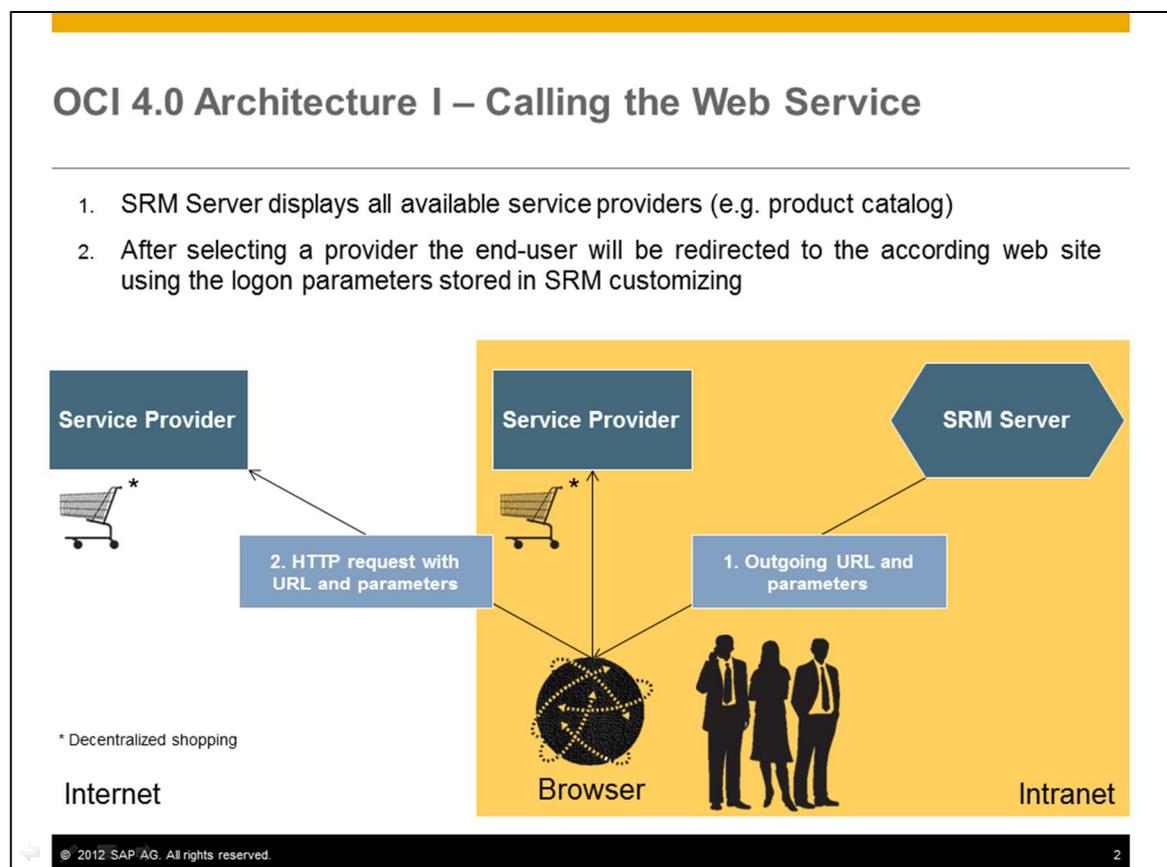


Figure 2 Calling the Web Service

The URL and the catalog parameters are defined in the Customizing for *SAP Supplier Relationship Management under Master Data* → *Define External Web Services (Catalogs, Vendor Lists etc.)*. In the technical settings, you can set GET or POST as the HTTP method for the call-up, the standard value is POST. The user's browser then calls up the catalog using the URL and the parameters.

Depending on the situation, other parameters besides those defined in Customizing for SAP Supplier Relationship Management can also be transferred to the catalog on call-up. These might be, for example, to provide the catalog with generic data or to activate specific functions in the catalog.

[Section 2.4](#) describes a summary of the transferred parameters.

2.1 Settings in the SRM Server

In the SRM Server, you must enter the URL and parameters of the catalog in the following order in the Customizing for *Supplier Relationship Management: SRM Server* → *Master Data* → *Define External Web Services (Catalogs, Vendor Lists etc.)*.

2.2 URL of the Product Catalog

You enter the URL of the catalog in the first line of the call structure; all subsequent entries (including the return URL in line 4) are transferred to the catalog as parameters. You do not need to specify a parameter name for the URL itself. *URL* is used here as the type. If the URL is longer than the field, you can distribute the URL over several successive fields that are then all of the type, *URL*. You cannot enter Parameters as part of the URL; maintain them separately as described in the following section.

2.2.1 Parameters

Following the URL, you must specify all parameters that the catalog requires on call-up. The provider of the catalog must have documented the names and valid values of these parameters.

The parameter type can be either *Fixed Value* or *SAP Field*. This parameter type determines how the value of the parameter is determined from the *Content* column. In the case of parameter type, *fixed values*, the system enters the value of the parameter directly in the *Content column*. In the case of parameters of type, *SAP Field* (generic parameters), the name of a SAP System variable exists. For the value of the parameter, the content of this system variable at runtime is used. This way it is possible, for example, to transfer the system language as a parameter, you choose the parameter type as *SAP Field* and *sy-langu* as content. In this manner, you can transfer all the globally available fields at runtime from the *SRM Server* as parameters.

Transfer of additional parameters

If the fixed values and the generic parameter values are not sufficient for the chosen process, you can implement the Business Add-In (BAdI), *Transfer Additional Parameters* (BBP_CAT_CALL_ENRICH) to transfer additional parameters from the SRM Server system to the catalog. Using this BAdI, you can determine and transfer multiple name-value pairs, to come up when you call a catalog. In addition, if you want to change the format of generic parameters, you can use this BAdI to convert these parameters. For example, you can use this BAdI to convert the system language to a different format: *DE- de* instead of simply *DE*.

As of SAP SRM 2.0, a sample implementation is available for the BAdI *Transfer Additional Parameters* (BBP_CAT_CALL_ENRICH), which reads the relevant user data in the SRM Server and transfers it to the catalog (however, the sample implementation is only run if the business type of the category is set to *E-form*).

As an example of how this data can be used by the catalog, see the ASP page (ASP = Active Server Pages) that is stored in the *SRM Server* System in the Internet Service BBP_FREEFORM in the file *freeform.zip*. When you open all the data in an executable directory on an ASP-compatible Web server, you can use this page as a catalog application for ordering business cards. All the form fields automatically have the default values that have previously been transferred by the BAdI implementation. The user only needs to check the accuracy of the entries.

This ASP page can, of course, only serve as a template. The example is intended to show the options that are available on a project basis with a catalog and the implementation of the BAdI *Transfer Additional Parameters* (BBP_CAT_CALL_ENRICH).

2.2.2 Return URL

The return URL is required so that the data from the catalog can be transferred back to the SRM Server application via the user's browser. The catalog must place it in the *Action* attribute of the transfer form. The return URL is also transferred to the catalog as a parameter. It usually contains other parameters (see below) that must first be extracted and placed in separate input fields (of type *hidden*) of the form.

All parameters after the return URL are generated as parameters for the return URL; they are not transferred to the catalog as individual parameters. You can name the parameter for the return URL as you wish since the parameter must be evaluated by the catalog (usually *HOOK_URL* is used as the parameter name). The value of the return URL must be empty; the actual return URL is determined during run time.

As of SRM Server 4.0, the specification of the return URL and the following parameters is optional in Customizing. If they are not specified, the return URL including the parameters is generated automatically. In this case *HOOK_URL* is used as the name for the return URL. The name of the return URL must be specified only if the catalog expects a different name for the return URL than *HOOK_URL*. Even if the name has been specified for the return URL, as of *SRM Server 4.0* the parameters for the return URL need no longer be specified; they are also generated automatically in this case.

2.3 Additional Functions in the Product Catalog

The product catalog can provide additional functions that can be used by *SRM Server* applications. In order to trigger these functions, additional parameters (defined within *OCI*) are transferred to the catalog when it is called up. If the product catalog supports one or more of the following functions, this is to be stated in the product catalog documentation. This is because an appropriate flag must be set in SRM Server Customizing so that the relevant SRM Server application can make use of the functionality. The fact that the interface supports the additional functionality does not necessarily mean that the functionality is actually used by a SRM Server application.

Currently the following functionality is supported by the *OCI*:

2.3.1 Detailed Display of a Product or Service

A product that has already been transferred to a SRM Server document using the *OCI* is to be displayed again later in the catalog. This serves to display detail data directly in the catalog that may not be defined in the *OCI* and that is also not available in the SRM Server.

In order to activate the function, the following parameters are transferred to the product catalog on call-up:

Name	Value
FUNCTION	DETAIL
PRODUCTID	<Database key for the product in the catalog>

Then the product catalog should immediately display the detail view of the corresponding product. With this function no data is transferred from the product catalog to the SRM Server.

A prerequisite for this function is that before the first time this product is transferred from the product catalog to the SRM Server System, the field *NEW_ITEM- EXT_PRODUCT_ID* is filled with the database key of the product in the catalog. In addition, in Customizing for *SAP Supplier Relationship Management* under *SRM Server -> Master Data -> Define External Web Services (Catalogs, Vendor Lists etc.)* for the corresponding catalog, the *Display Product Data Again in Catalog* checkbox must be selected.

2.3.2 Validation of a Product

If an item from a product catalog has been added to a template for an SAP SRM document then it is possible, for example, that the price of the product has changed over time. If a new SAP SRM document were to be created from this template, the price change would usually not be taken into account. The function serves to update the product information in the template while the SAP SRM document is being created.

In order to activate the function, the following parameters are transferred to the product catalog on call-up:

Name	Value
------	-------

FUNCTION	VALIDATE
PRODUCTID	<Database key for the product in the catalog>
QUANTITY	<Current purchase order quantity>

The parameter QUANTITY is transferred as of OCI 3.0 so that the catalog can determine the correct price from a scale, if appropriate.

Then the product catalog replies with an HTML page that contains a form with the product data in OCI format. The return URL must be split here as described in Sections [2.2.2](#) and [3.1](#). The HTML page may not contain any visible elements (the input fields must be of the type *hidden*). The form must be sent automatically by JavaScript after the page has been loaded.

A prerequisite for this function is that before the first time this product is transferred from the product catalog to the SRM Server System, the field NEW_ITEM- EXT_PRODUCT_ID is filled with the database key of the product in the catalog. In addition, in SRM Server Customizing the *Validate Product Data from SAP Enterprise Buyer* checkbox must be selected for the relevant catalog.

2.3.3 Sourcing/Product Search

In order to activate the function, the following parameters are transferred to the product catalog on call-up:

Name	Value
FUNCTION	SOURCING
SEARCHSTRING	<Search term>
VENDOR	<Business partner number in EBP>

The catalog replies with the display of the search results that correspond to the specified parameters. By navigating further in the product catalog, you can select items as normal and transfer them to the SRM Server application.

This function has not yet been implemented in the SRM Server.

2.3.4 Background Search

In order to avoid the situation where a user searching for a particular product has to search multiple catalogs in sequence, a Cross-Catalog Search has been introduced as of SRM 3.0. The user enters a search term once in the SRM Server application and this term is transmitted to all catalogs that support the function when the catalogs are called up. The SRM Server displays the search results from all catalogs in a list and the user can select individual products from here.

In order to activate the function, the following parameters are transferred to the product catalog:

Name	Value
FUNCTION	BACKGROUND_SEARCH
SEARCHSTRING	<Search term>

The product catalog replies with a HTML page that contains the search results in OCI format. All search results must be accessible on this page without having to use a scroll function since the page is evaluated automatically. The form from this page may not be sent automatically to the return URL (in contrast to validation of a product).

A prerequisite for this function is that the *Cross-Catalog Search* checkbox is selected in SRM Server Customizing for the relevant catalog.

2.4 Overview of the Call-Up Parameters

The following parameters are transferred when the catalog is called up using either HTTP

GET or POST, depending on what you define in SRM Server Customizing:

Description	Name of the Parameter	Content or Example
Parameters from <i>SRM Server Customizing</i>	As specified in <i>SRM Server Customizing</i>	See Section 2.2.1
Parameters from a BAdI implementation	As implemented in the BAdI	See Section 2.2.1
Parameters to trigger additional functions	As described in Additional Functions in the Product Catalog	See Section 2.3
Return URL	If not specified differently in <i>SRM Server Customizing</i> : ,HOOK_URL'	URL with parameters in the query string See Section 2.2.2
Interface version <i>OCI</i>	OCI_VERSION	For example: 4.0
Interface version <i>OPI</i>	OPI_VERSION	For example: 1.0
Character set of <i>SRM Server</i> application	http_content_charset	For example: iso-8859-1
Target (HTML target) for return to the <i>SRM Server</i> application	returntarget	For example: _parent

3 Return from Catalog

An HTML form is used to transfer the selected product data to the SRM Server. This form is part of a HTML page that must be created by the catalog. This page (the last page that is displayed by the catalog) is sent to the user's browser. The user can now send the form from this page to the SRM Server application that then takes over the form data.

OCI 4.0 Architecture II – Returning data to SRM Server

1. Service provider creates an HTML page containing the OCI compliant data
2. End-user passes this data on to SRM Server for further processing
3. SRM Server receives and processes the incoming data from the web service

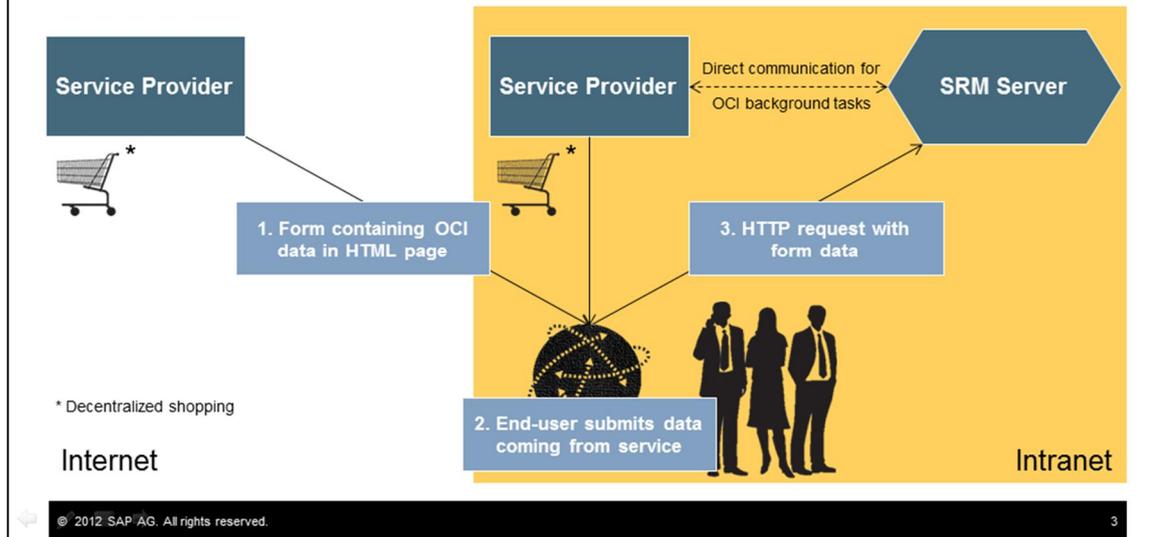


Figure 3 Returning data to SRM Server

In order for you to transfer the data from the catalog to the SRM Server using the user's browser, the return URL described in [Section 2.2.2](#) and [3.1](#) must be inserted into the *Action* attribute of the HTML form created by the catalog (the *Action* attribute of the form is the URL to which the form data is sent).

The data to be transferred is transported within the input fields of the form; the field names must adhere to the syntax specified below. The *Type* attribute of the input fields should be *text* or *hidden*. The HTTP method recommended in all cases is POST because using GET can lead to browser-dependent length restrictions.

3.1 Use of the Call-Up Parameters from Section 2.4

- Use of the parameters from SAP SRM Customizing is catalog-specific; here the only parameters transferred are those that the catalog expects.
- Use of the parameters from a possible BAdI implementation is project-specific; here the only parameters transferred are those that the catalog expects in the context of a customer project.
- The expected reactions to the parameters for activating additional functions have already been described in [Section 2.3](#).
- The interface version has purely informative character; it clarifies which fields or processes are already supported by the SRM Server application.
- The return URL points to the current SAP SRM application. It also contains further parameters.
- The URL without these parameters must be placed by the catalog into the *action* attribute of the transfer form. The parameter names and their values must be placed into the attributes *name* and *value* of the individual *input* fields of the transfer form; the fields are to be of the type *hidden*.
- The character set used by the SAP SRM application is transferred to the catalog (parameter name: *http_content_charset*). When generating the HTML page with the transfer form, the catalog must use this character set and must insert it into the *meta*

tag after the detail *charset=*.

- The target for the return to the SRM Server application (parameter name: *returntarget*) must be inserted into the *target* attribute of the transfer form.

3.2 Fields and Field Checks

The naming convention for the fields in the OCI is as follows:

NEW_ITEM-<Field name>[<index>]. The field type is always CHAR.

Field name	Length	Description
NEW_ITEM-DESCRIPTION[n]	40	Description of the item
NEW_ITEM-MATNR[n]	40	SRM product number of the item
NEW_ITEM-QUANTITY[n]	15	Item quantity (1.)
NEW_ITEM-UNIT[n]	3	Quantity unit for item quantity (3.)
NEW_ITEM-PRICE[n]	15	Price of an item per price unit (1.)
NEW_ITEM-CURRENCY[n]	5	Item currency (3.)
NEW_ITEM-PRICEUNIT[n]	5	Price unit of the item (if empty, 1 is used)(2.)
NEW_ITEM-LEADTIME[n]	5	Delivery time of the item in days (2.)
NEW_ITEM-LONGTEXT_n:132[]	∞	Long text for the item (4.)
NEW_ITEM-VENDOR[n]	10	SRM vendor number (business partner) for the item
NEW_ITEM-VENDORMAT[n]	40	Vendor product number for the item
NEW_ITEM-MANUFACTCODE[n]	10	SRM manufacturer number of the item
NEW_ITEM-MANUFACTMAT[n]	40	Item's manufacturer part number
NEW_ITEM-MATGROUP[n]	10	SRM material group for the item
NEW_ITEM-SERVICE[n]	1	Flag: the item is a service.
NEW_ITEM-CONTRACT[n]	10	SRM contract to which the item refers
NEW_ITEM-CONTRACT_ITEM[n]	5	Item within the SRM contract
NEW_ITEM-EXT_QUOTE_ID[n]	35	Number of an external bid for this item (as reference for a subsequent purchase order)
NEW_ITEM-EXT_QUOTE_ITEM[n]	10	Item of external bid
NEW_ITEM-EXT_PRODUCT_ID[n]	40	Unique database key for this item in the catalog
NEW_ITEM-ATTACHMENT[n]	255	URL of the attachment (the attachment must be accessible for downloading under this URL)
NEW_ITEM-ATTACHMENT_TITLE[n]	255	Title of the attachment (if this is empty the file name from the URL above is used)
NEW_ITEM-ATTACHMENT_PURPOSE[n]	1	Purpose of the attachment. C corresponds here to configuration.
NEW_ITEM-EXT_SCHEMA_TYPE[n]	10	Name of a schema via which it was imported in the SRM Server
NEW_ITEM-EXT_CATEGORY_ID[n]	60	Unique key for an external category from the schema above, independent of the version of the schema

NEW_ITEM-EXT_CATEGORY[n]	40	Unique key for an external category from the schema above, dependent on the version of the schema
NEW_ITEM-SLD_SYS_NAME[n]	60	Name of a system in the System Landscape Directory (SLD)
NEW_ITEM-CUST_FIELD1[n]	10	User-defined field
NEW_ITEM-CUST_FIELD2[n]	10	User-defined field
NEW_ITEM-CUST_FIELD3[n]	10	User-defined field
NEW_ITEM-CUST_FIELD4[n]	20	User-defined field
NEW_ITEM-CUST_FIELD5[n]	50	User-defined field
NEW_ITEM-ITEM_TYPE[n]	1	Item type Root, Outline, Leaf. Introduced since SRM 7.0 onwards.
NEW_ITEM-PARENT_ID[n]	5	An integer field that defines what the parent of the current item is in the Hierarchy structure. This field will be blank for an item which does not have any parent. Introduced since SRM 7.0 onwards.

- 1.) 11 digits before the decimal point, 3 after it. Do not use commas for thousands.
- 2.) In whole numbers.
- 3.) Must be maintained as ISO code in the *SRM Server*.
- 4.) The field NEW_ITEM-LONGTEXT_n:132[] is an exception as far as the syntax of the index n is concerned.
The field length is unlimited.

3.2.1 Required and Optional Fields

The following fields are required fields in all cases:

- Either NEW_ITEM-DESCRIPTION[n] or NEW_ITEM-MATNR[n] must be filled. Only one of the two should be filled.
- NEW_ITEM-QUANTITY[n]

The following fields are required fields depending on conditions:

- NEW_ITEM-UNIT[n] if NEW_ITEM-MATNR[n] has not been filled
- NEW_ITEM-CURRENCY[n] if NEW_ITEM-PRICE[n] has been filled
- NEW_ITEM-EXT_SCHEMA_TYPE[n] if NEW_ITEM-EXT_CATEGORY_ID[n] or NEW_ITEM-EXT_CATEGORY[n] are used
- NEW_ITEM-EXT_QUOTE_ID[n] if NEW_ITEM-EXT_QUOTE_ITEM[n] has been used
- NEW_ITEM-CONTRACT[n] if NEW_ITEM-CONTRACT_ITEM[n] has been used

All other fields are optional.

3.2.2 Product Numbers

There are four fields in the interface that describe product numbers:

- NEW_ITEM-MATNR[n]: The product number in the SRM System of the purchaser
- NEW_ITEM-VENDORMAT[n]: The supplier's product number
- NEW_ITEM-MANUFACTMAT[n]: The manufacturer's product number

- **NEW_ITEM-EXT_PRODUCT_ID[n]**: The number that uniquely identifies the product in the catalog.

These product numbers must not be mixed or used for other purposes; in particular the field **NEW_ITEM-MATNR[n]** must only be filled if the product number in the customer system is known to the catalog.

3.2.3 Configurable Products

Some products (such as PCs) can be configured in the catalog. However, the configuration information is not part of the OCI since the structure of this information differs greatly between providers. There are three options for transferring such products with the OCI without losing the configuration information.

- The catalog can create a bid in the sales system and can store the configuration information there. It can then use the fields **NEW_ITEM-EXT_QUOTE_ID[n]** and **NEW_ITEM-EXT_QUOTE_ITEM[n]** to transfer a reference to the bid. The bid number is copied to the SRM Server. The configuration information is only available in the sales system if you use this option. This variant is suitable for the local and extended classic scenario since the bid reference is not transferred to MM backend systems as standard. If, however, you wish the bid reference to be transferred, you can copy it in **BA01 BBP_CATALOG_TRANSFER** into the purchase order text for the item.
- The field **NEW_ITEM-LONGTEXT_n:132[]** can be used to transfer the configuration information as text. The content of the field is included in the purchase order text of the SRM Server shopping cart and of the subsequent purchase order; this way the configuration information is available in the SRM Server.
- The fields **NEW_ITEM-ATTACHMENT[n]** and **NEW_ITEM-ATTACHMENT_PURPOSE[n]** can be used to transport the configuration information. Since you can transfer files of any type as attachments, you should ensure that the file can also be displayed (using proprietary or uncommon file types is therefore not recommended). If you use XML files, for example, you should ensure that the formatting information (XSLT) is also included so that the file can be displayed. The configuration information is also available in the *SRM Server* with this alternative. This variant is only suitable for the local and the extended classic scenario because attachments are not currently transferred to MM backend systems.

3.2.4 External Product Categories

As of SAP Enterprise Buyer 3.0, product categories from external schemas can also be used in the OCI. However, a prerequisite is the import of the relevant schema using the Product Content Workbench (PCW). If the imported schema is also used internally in the SRM Server, then no further action is required. If, however, there is to be a link from the product category from the external schema to an internal product category in the SRM Server, the mapping of the imported (or parked) schema must first be completed using the PCW. The fields **NEW_ITEM-EXT_SCHEMA_TYPE[n]** and **NEW_ITEM-EXT_CATEGORY_ID[n]** are used for the external product categories.

If, for example, an UNSPSC schema has been imported into the SRM Server under the name *UNSPSC_Vendor1*, then *UNSPSC_Vendor1* is the value that must be transferred in the field **NEW_ITEM-EXT_SCHEMA_TYPE[n]**.

3.2.5 Introduction of New fields to enable transfer of Item Hierarchy to SRM

To accommodate the transfer of item hierarchy information from the catalog to SAP SRM, the following new fields have been added to the current OCI interface. This function is supported from SRM 7.0 onwards.

Field Name	Length	Description
------------	--------	-------------

NEW_ITEM-PARENT_ID[n]	5	An integer field that defines what the parent of the current item is in the hierarchy structure. This field will be blank for a parent/normal item.
NEW_ITEM-ITEM_TYPE[n]	1	A character field that defines the type of the current item is in the hierarchy structure.

The above addition will ensure that each item is transferred along with its parent's ID and item type to SAP SRM.

Possible values for the Field PARENT_ID

- For hierarchical items which are the top most parent node: The parent field is blank.
- For hierarchical items which are children of other items: The parent field is set to the immediate parent in the hierarchical structure with the ID of the parent line item.
- For regular items, which are neither parent nor child, the transfer will happen in the same way as it happened previously. In this case the PARENT_ID field is blank.

Possible values for the Field ITEM_TYPE

- If the item is at the top level of the hierarchy, then the value will be "R"
- If the item is an outline, then the value will be "O"
- If the item is a functional line item, then the value will be "F"

3.2.6 Introduction of Inbound Handler and Outbound Handler

Previously., the catalog was launched directly from SAP SRM and the item data was passed to SAP SRM directly from the catalog.

From SAP SRM 6.0 onwards, SAP SRM no longer uses ITS technology and several features which were previously supported by ITS were missing, including code page conversion, encoding and decoding HTML tags, launching catalog in an external popup window without ending the SRM session. To include these features, SAP SRM introduced two new services called inbound handler and outbound handler.

Outbound handler is called while launching the catalog. This displays a *Back to Application* link at the top of every catalog page to allow user to navigate back to SAP SRM without shopping for any items.

Inbound handler is called while transferring the items back from catalog. It is involved in code page conversion and encoding and decoding the HTML tags present in the OCI data.

These services are very useful but if customers want to improve the performance of the system they can switch off these services by adding some new entries in the catalog standard call structure in transaction SPRO.

These new entries are described below:

Parameter Name	Type	Remarks
BYPASS_OUTB_HANDLER	Fixed Value	This entry is needed to switch on or off the outbound handler. By default, the outbound handler is switched on. If the entry is there in the standard call structure and the value is "X", then outbound handler will be switched off.

BYPASS_INB_HANDLER	Fixed Value	This entry is needed to switch on or off the inbound handler. By default the inbound handler is switched on. If the entry is there in the standard call structure and the value is "X", then inbound handler will be switched off.
--------------------	-------------	--

3.2.7 Introduction of Portal Navigation while transferring data to SRM

Changes have been made to the previous approach to data transfer processes between the catalog and SAP SRM. Previously, catalog posted a blank HTML form with OCI data in the hidden input fields and SAP SRM read the data and displayed it in the shopping cart or in other business objects.

To activate this portal navigation, customer must make a new entry in the standard call structure of the Catalog.

Parameter Name	Type	Remarks
usePortalNavigation	Fixed Value	This entry is needed to enable portal navigation for catalog. If the entry is present in the standard call structure and the value is X, then catalog will use Portal Navigation API to transfer the items back to SRM. If this is turned on, Inbound Handler and Outbound Handler is automatically turned off.

3.2.8 Customer-Specific Extensions in the OCI

To transfer additional fields in the OCI, there are two options:

Using the Fields NEW_ITEM-CUSTFIELD1-5

These fields have no fixed semantics in the OCI and this means that you can transfer any contents in order to then evaluate them in BAdI BBP_CATALOG_TRANSFER. In releases prior to SAP Enterprise Buyer 3.0 it is not possible to transfer these fields directly to the shopping cart item. If the content is to be reused later, it must be saved in the BAdI above in the customer's own database table.

As of SAP Enterprise Buyer 3.0, it is possible to include these fields (without the prefix NEW_ITEM-) in the customer-specific Include CI_BBP_ITEM_SC; the contents are then automatically forwarded to the shopping cart and saved there.

Using the Include CI_OCI_CUSTOMER_EXTENSION

This customer-specific Include is available as of SAP Enterprise Buyer 3.0; it allows you to extend the OCI generically. All fields that are created in the Include can then be transferred by the catalog as *NEW_ITEM-<field name from Include>[n]*.

The fields are also available in the BAdI BBP_CATALOG_TRANSFER. If you require the fields to also be available in the shopping cart and saved there, you must also create them in the Include CI_BBP_ITEM_SC.

For more information on the use of customer-specific Includes in SRM Server, see SAP Note 458591.

3.3 XML Variant of the OCI

The OCI can also process an XML file. Here, the same architecture is used as in the pure HTML variant, this means that the XML data is embedded in an HTML form for the transfer from the catalog to the SRM Server using the user's browser.

To transfer an XML file, aside from the fields mentioned in [Section 2.3](#), two further HTML-inputfields are used:

xmltype:

This parameter specifies the type of the XML file used so that the correct XML mapping can be found. Up to and including SAP Enterprise Buyer 3.0 the mapping of the received XML

data is done exclusively in the Business Connector, which must be set up for this purpose for the relevant SAP Enterprise Buyer System.

As of SAP Enterprise Buyer 3.5, the mapping can also be done in the SRM Server itself. A prerequisite is that the corresponding XML schema is used. The previous schemas are also supported.

Valid values for the field type *xmlType* as of *SRM 3.0* are:

Value	Description	DTD/Schema/Mapping
ESAPO	Encoded SAP Object for <i>OCI</i> Version up to and including 3.0	PDI_OCI.dtd/PDI_OCI.xsd/BC
ESAPO3.0	Encoded SAP Object for <i>OCI</i> Version up to and including 3.0	PDI_OCI_30.dtd/PDI_OCI_30.xsd/BC
ESAPO3.5	Encoded SAP Object for <i>OCI</i> Version as of 3.5	OpenCatalogInterface.xsd/im <i>SRM Server</i> .

~xmlDocument:

In this parameter, the XML file that must correspond to one of the schemas above is transferred as a Base64-coded character set. The coding can be done either directly on the server page of the catalog or, as shown in the sample application, by JavaScript on the client's page.

4 Handling of the Browser Window

The catalog is displayed in a separate browser window. This window is both opened and closed using the SRM Server application.

The HTTP response of the SRM Server application to the HTTP request with the OCI data consists of an HTML page that contains script for refreshing the window with the SRM Server application and for subsequently closing the catalog window. This page assumes that the JavaScript reference window.opener points to the browser window of the SRM Server application.

You must not change the name of the original window object because this would destroy the reference between the SRM Server application window and the catalog window. If this were to happen, the SRM Server application would no longer be able to correctly display the data that has been transferred from the catalog or close the catalog window.

If the catalog opens additional windows during the search, it must close these before the data is transferred back to the SRM Server.

5 Secure OCI

As of OCI 4.0, it is now possible to integrate the catalog in such a way that sensitive data including username, password, price, is no longer sent to the catalog over the browser. This prevents unauthorized dissemination of confidential login information and modification of price with mischievous intentions. Confidential information can now be exchanged by the catalog over a backend HTTP call, which is safe from prying eyes and unwanted modifications.

5.1 Launching Catalog with Secure OCI

When using secure OCI, the launch of catalog application does not happen in a single call from SRM Server to catalog application. There are two calls sent from SAP SRM to the catalog application:

1. Backend call to retrieve the IDs.
2. Request to launch the catalog application using one of the IDs.

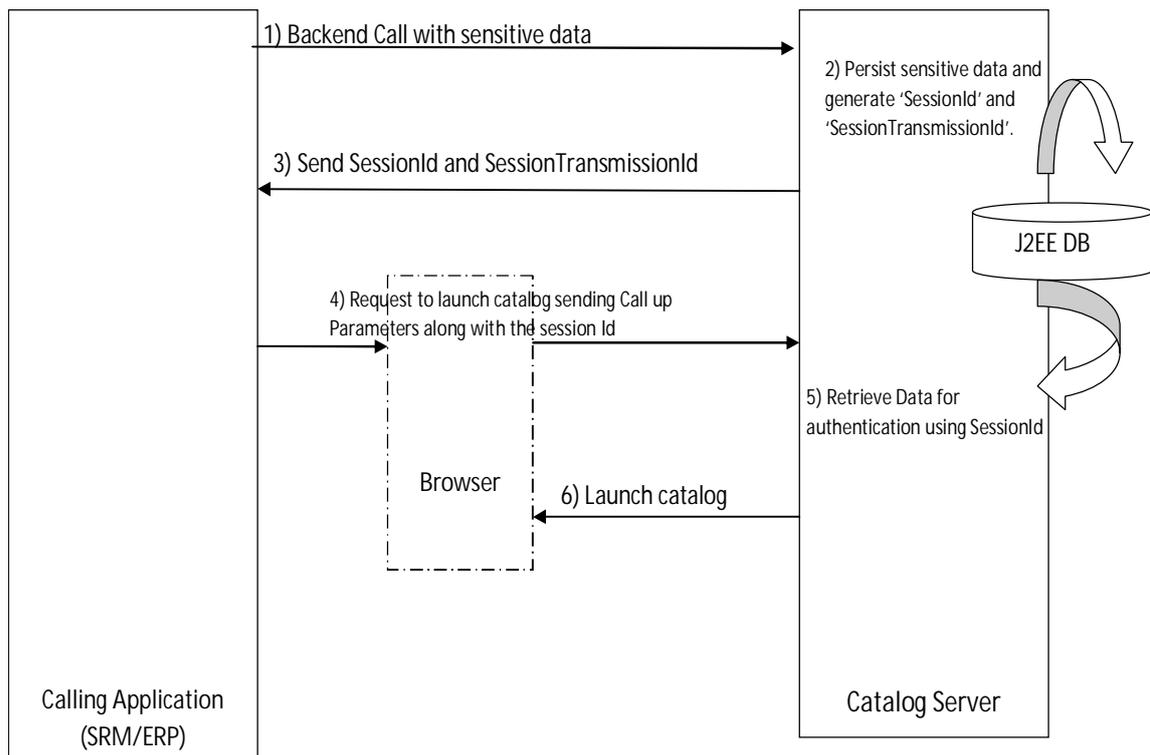


Figure 4 Launch catalog through Secure OCI

5.1.1 Initial Handshake

If Secure OCI is inactive, the call is generated as mentioned in [Section 2](#). Otherwise, the OCI must check whether the following parameters are provided in the request or not:

Here is the list of parameters and their possible values required to use Secure OCI:

PARAMETER NAME	Description	VALUE	TYPE
SECURE_AUTH_URL	Contains the URL of the web service or servlet that handles the backend call from OCI	URL of the web service or servlet.	Fixed Value
SECURE_TRANSMISSION	Indicates the call for retrieving session data	X	Fixed Value
FUNCTION	Indicates the action to be taken by the catalog application, in this case generate 'SessionId' and 'SessionTransmissionId'	INITIALIZE	Fixed Value
SECURE_AUTH_METHOD	Determines the HTTP method to be used to send response to the backend call	GET/POST Default to POST	Fixed Value

A backend call is generated using the following parameters along with the other parameters such as username and password.

5.1.2 Response to backend call

In response to the backend call, the catalog application sends an XML. The following XML will be sent as a response to the initialization backend call:

```
<SecureOCI>
```

<SessionId>8081457345719230120</SessionId>

<SessionTransmissionId>7551489217863242754</SessionTransmissionId>

</SecureOCI>

These elements along with username and password need to be persisted for further communication.

SessionId: This should be sent as an URL parameter to launch the catalog.

SessionTransmissionId: This parameter will be used in the OCI backend call structure that will be sent when the control returns from the catalog application to retrieve the checked out item data.



NOTE

The 'SessionId' and 'SessionTransmissionId' can be used only once per session. Once the items are transferred to SAP SRM, the entry corresponding to these IDs must be deleted from DB.

5.2 Return from catalog

In the absence of Secure OCI, when the control returns from catalog, the item data is directly sent as an HTML form. When Secure OCI is enabled, the catalog application will not send the item data in the response. In order to fetch item data from the catalog application, the calling application has to initiate a backend call.

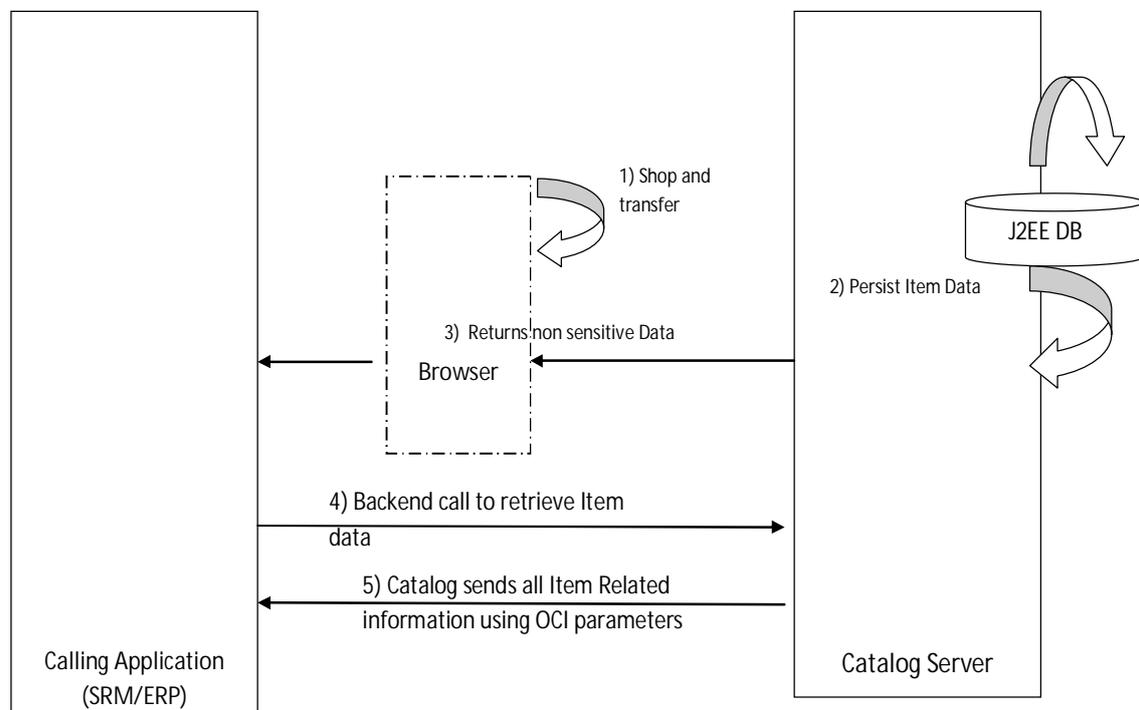


Figure 5 Return from catalog through Secure OCI

5.2.1 Initiate backend call

The OCI has to initiate the backend call to the catalog application by including the following parameters in its call structure:

Here is the list of above mentioned parameters and their possible values required to use Secure OCI:

PARAMETER NAME	Description	VALUE	TYPE
SessionTransmissionId	Used in the backend call to retrieve the checked out item data	7551489217863242754	
FUNCTION	Indicates the action to be taken by the catalog application, in this case send the item data	'RETRIEVEOCI'	Fixed Value
SECURE_AUTH_METHOD	Determines the HTTP method to be used to send response to the backend call	POST	Fixed Value

5.3 Enablement of Secure OCI for catalogs

For the implementation details, refer to the SAP Technical Note - 1574737.

6 Enhancements in OCI 5.0 for Mass Upload

With OCI 5.0, data from catalogs can be replicated to the SRM Server and to SAP Enterprise Search to enable a federated search. End users can then directly search for catalog items within the SAP SRM shopping cart application, view the item details, and add them to the shopping cart directly. These enhancements are currently only available to customers of SAP SRM who have also installed the SAP SRM User Interface Add-On.

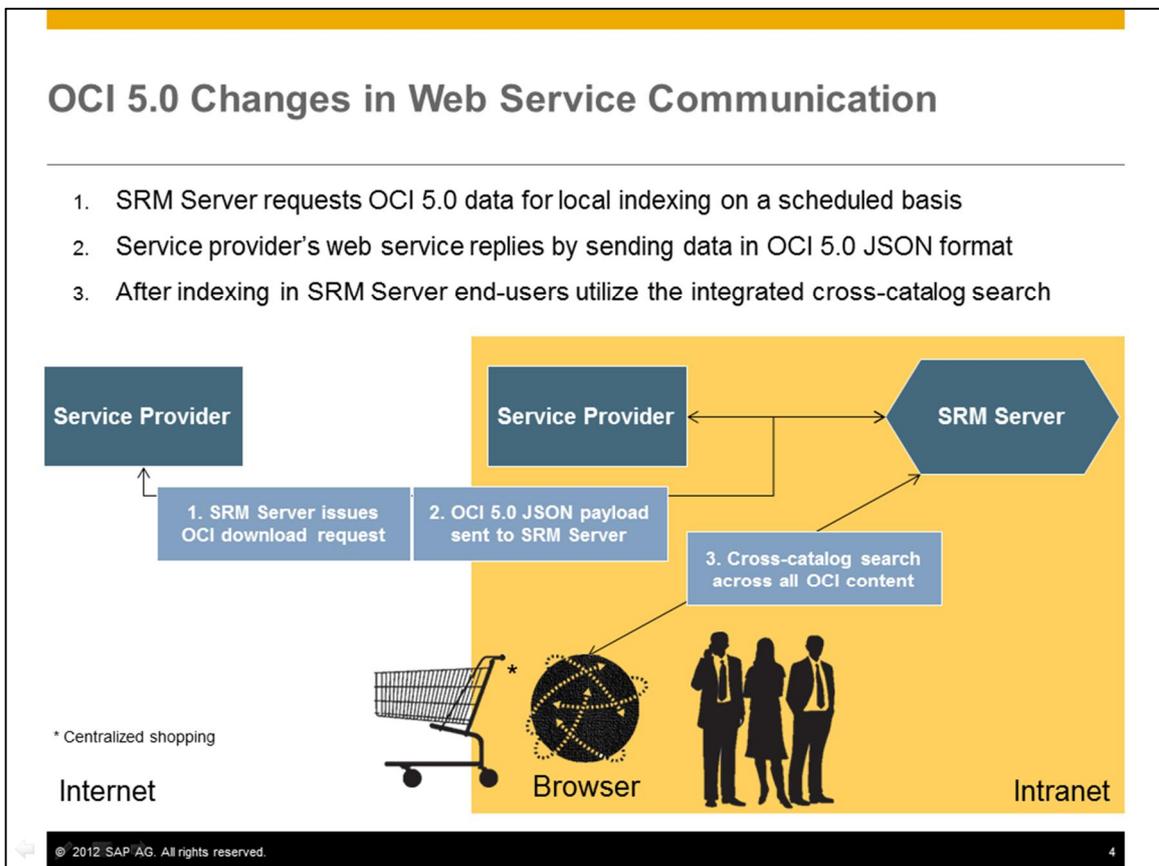


Figure 6 OCI 5.0 changes in Web Service Communication



NOTE

The search function uses SAP Enterprise Search and TREX. For more information on configuring SAP Enterprise Search & TREX, please see SAP Help documentation on http://help.sap.com/saphelp_nw70ehp3/helpdata/en/48/72305df518055ee1000000a42189b/frameset.htm



NOTE

Images are stored in SAP Content Management System (SAP CMS) under the content repository SRMNXP_IMG. For information on how to configure SAP CMS, please see SAP Help documentation on http://help.sap.com/saphelp_nw70ehp3/helpdata/en/1f/2df44eeebf8145b67b31f513a83fc5/frameset.htm

There are multiple ways by which data can be mass replicated to the SRM System.

- JSON (Java Script Object Notation) from a physical file as explained in [Section 6.1](#)
- JSON over HTTP (Hyper Text Transfer Protocol) as explained in [Section 6.2](#)

Catalogs have the option of sending only partial data from their catalogs for mass replication. This may be required in scenarios where the price or other important attributes change too often (for example, the prices for crude oil can change every hour). In such scenarios, the catalog can choose not to send the price data. Catalog should send value X for the parameter CATALOG_MANAGED . The item will appear in the search results, but it can be bought only from the catalog. The end user will see the button “Go to Catalog” instead of “Add to Cart” for such items in their search results. This is explained further in [Section 6.7.1](#).

6.1 Mass replication of data delivered in a physical file

This option allows mass replication of data using physical files in Java Script Object Notation (JSON) format. Catalogs can deliver a JSON file along with images. The images can be referring to a http(s) URL or can be provided in a physical file placed in the application server (if indexing is scheduled in a background task). In case of images being provided in a physical file all the images should be in one folder and not contained within subfolders. The names of the images should match the names specified in the JSON file. HTTP based transfer of attachments is recommended over physical upload of images from a folder location. Details regarding attachment upload are provided in [Section 6.3](#).

6.2 Mass replication of data over HTTP(s) connection

This option allows for mass replication of data in Java Script Object Notation (JSON) format sent over an HTTP(S) connection. Catalog providers must provide an HTTP(S) URL through which the data can be pulled from the catalog server in JSON format. Attachments should be delivered as an HTTP(S) URI or as a physical folder on the application server. The names of the attachments should match the names specified in the JSON data. All attachments should be in one folder and not contained within subfolders. HTTP based transfer of attachments is recommended over physical upload of images from a folder location. Details regarding attachment upload are provided in [Section 6.3](#)

6.3 Replicating Attachments

There are three ways of making item attachments available to end-users in the SRM NXP Shopping Cart screen.

- Provide the physical files of the attachments and store them in a server location as a folder or a zip file. For example, an attachment named *laptop.jpg* is stored in a folder

named *Catalog Attachments*. In the JSON file, for the item *laptop* in the object attachments *laptop.jpg* is specified as the attachment name with the attachment type 01 identifying it as the main image. Then the system would read it and upload it to SAP Content Management System (CMS) against the item *laptop*.

- Provide URL for an attachment, which can then either be:
 1. Downloaded from the URL and uploaded into SAP CMS. When the user views this item, the attachments are served from SAP CMS.
 2. URL is stored without any changes. When the user views the associated catalog item, the attachments are pulled directly from the URL by the browser.



CAUTION

If you are running your SRM Server instance in HTTPS (Secure HTTP) mode, and the attachment URL is in HTTP (standard HTTP) mode, end users may get a warning that the page is accessing unsecure resources. If the attachments are being retrieved from a URL located on the internet, then your users will need access to the internet.



NOTE

Reading and uploading attachments takes time irrespective of whether they are read from a URL location on the internet or from file system. If your data has lot of attachments, the upload report/job can take a long time to finish..

6.4 Format for Data Transfer

Data transfer for mass upload in OCI 5.0 happens in JSON format. This enables high volume data processing with high performance. The rules and complete details of the format are specified in [Section 6.8](#).

To indicate a delta or a full transfer of the catalog items, the Complete Transmission Indicator has to be sent as part of the first JSON response. In case of manual file upload, the system administrator chooses whether the upload is a full upload or a delta upload.

6.5 Settings in the SRM Server

In this section, settings required on the SRM Server are described.

6.5.1 Define External Web Services

For authorization management and management of catalog data visibility between different Purchasing Groups and Organizations, SAP SRM customers are required to create a logical entry for each catalog in the Customizing for *Supplier Relationship Management under SRM Server* → *Master Data* → *Define External Web Services (Catalogs, Vendor Lists etc.)*. You can find the complete documentation in the description of the Customizing *Define External Web Services (Catalogs, Vendor Lists etc.)*.

6.5.2 Activate SICF service for delivering images/attachments

The service to fetch images and attachments can be activated in SAP SRM either in transaction *SPRO*, navigate to Customizing for *Supplier Relationship Management -> SAP SRM User Interface Add-on -> Activate ICF Service for Image Delivery*, or in transaction *SICF*, navigate to

the service, *Default_host* -> *sap* -> *srmnxp* -> *attachment loader* then right-click on this service and activate it.

6.5.3 Proxy server configuration in the BAdI SRM_GET_PROXY_INFO

If your organization mandates usage of a proxy server for accessing resources over the internet and you have selected the *Download attachments from their HTTP/HTTPS location* checkbox in the data upload report and the JSON data file you are uploading has documents (images/attachments) accessible over the internet, then for downloading such attachments you must specify the proxy server configuration to be used for accessing the internet in BAdI

SRM_CAT_GETPROXYINFO. You can find more documentation on the relevant configuration required in transaction SPRO, navigate to the Customizing for - *Supplier Relationship Management* under *SRM Server* -> *Customer Extension* -> *Business Add-ins* -> *Proxy Server Configuration*.

6.5.4 Customer Enhancements

You can enrich the existing data, by using the following two BAdIs:

- */SRMNX/CATALOG_TRANSFER_DATA*: You can use this BAdI for data enrichment before writing to staging tables in SRM for indexing.
- */SRMNX/CATALOG_ENRICH_DATA*: You can use this BAdI for data enrichment before adding the catalog item from search results to SRM Shopping Cart.

6.5.5 Additional settings for Mass Upload over Http(s)

6.5.5.1 URL of the Product Catalog

You have to enter the URL of the catalog in the first line of the call structure in the Customizing, *Define External Web Services*; all subsequent entries are transferred to the catalog as parameters. You do not need to specify a parameter name for the URL itself. *URL* is used here as the type. If the URL is longer than the field, you can distribute the URL over several successive fields that are then all of type *URL*. You cannot enter Parameters as part of the URL; maintain them separately as described in the following section.

6.5.5.2 Parameters

Following the URL, you must specify all parameters that the catalog requires on call-up. The provider of the catalog must have documented the names and valid values of these parameters.

The parameter type can be either *fixed value* or *SAP Field*. This parameter type determines how the value of the parameter is determined from the column *Content*. In the case of fixed values, the value of the parameter is entered directly in the column *Content*. In the case of parameters of type *SAP Field* (generic parameters), however, the name of a SAP System variable is there. For the value of the parameter, the content of this system variable at runtime is used. This way it is possible, for example, to transfer the system language as a parameter (you choose *SAP Field* as type and *sy-langu* as content). In this manner, you can transfer all the globally available fields at runtime from the SRM Server as parameters.

For the parameters required to support Mass upload over Http(s) Function and PageSize parameters (explained in below table) have to be specified.

URL Parameter Name	Value
FUNCTION	DOWNLOADJSON

PAGESIZE	Size of the page is the number of items which would be received in one single call. Default is 20. Customer can override the default value in the catalog upload cockpit.
----------	---

Transfer of additional parameters

If the fixed values and the generic parameter values are not sufficient for the chosen process, you can implement the Business Add-In (BAI) *Transfer Additional Parameters* (BBP_CAT_CALL_ENRICH) to transfer additional parameters from SRM Server System to the catalog. In this BAI, you can determine multiple name-value pairs that are then transferred to the catalog, when it is called up. Also, if the format of generic parameters is to be changed (for example, if the system language is to be transferred in a different format: *DE-de* instead of simply *DE*), you can use this BAI to convert these parameters.

6.6 Function for mass upload over Http(s)

Catalogs supporting the JSON download functionality over HTTP/HTTPS need to provide the following functionalities:

- Break up data into pages
- Serve a request for a particular page

6.6.1 Technical Specification for JSON download

In order to trigger the mass upload in JSON format the following URL parameters are transferred to the catalog on call-up:

URL Parameter Name	Value
FUNCTION	DOWNLOADJSON
PAGESIZE	Size of the page is the number of items, which would be received in one single call. Default is 20. Customer can override the default value in the catalog upload cockpit.
REQUESTEDPAGE	The page requested by the SRM server will be blank in the first call. The catalog should send always the first page, if this parameter is blank.
LASTUPDATED	Timestamp in DDMMYYYYHHMMSS format indicating the time where the last successful upload from the catalog happened. In case of initial load this field would be blank.
TRANSACTIONID	Optional TransactionID, which can be transferred by the catalog in the first response, the same TransactionID if exchanged during the first response would then be sent back during the rest of the calls.

The catalog returns the data in JSON format. As part of the JSON response it has to send the following parameters as part of the body.

Name	Value
CURRENTPAGE	The current page sent by the catalog in JSON format.
TOTALPAGES	The total number of pages the data will be delivered in.

TOTALITEMS	The total number of items for the given <i>From</i> and <i>To</i> timestamp.
TRANSACTIONID	A unique Identifier provided by the Catalog Provider on the first request for download and will be during the subsequent requests.
CTI	Complete Transmission Indicator, indicating that the entire data from the Catalog will be transferred. If not set to true would indicate only Delta transfer. This should be sent as part of the first request response



NOTE

The purpose of introducing the paging concept is to prevent timeout of HTTP/HTTPS calls and to enable more fine grained control over server loads, and network bandwidth usage. It also helps in case if the transfer fails for a particular page. In such a case it would not be required to pull all the preceding pages, which have already been successfully transferred and saved.



CAUTION

Since the above URL calls are stateless in nature, it is imperative that the catalog result should be consistent and repeatable for any given set of parameters. This also means that the sorting of the data should always be fixed even across multiple HTTP calls, otherwise data transfer would be inconsistent. The TRANSACTIONID could be used for the same.

An example of the return data format is given below:

```
{
  "CURRENTPAGE" : 1
  "TOTALPAGES" : 101
  "TOTALITEMS" : 2002,
  "TRANSACTIONID": 12323231,
  "CTI" : "X",
  Items :[
    {"NEWITEM":{
      "EXT_PRODUCT_ID": "1297767###62434_0001",
      "CATALOG_NAME": "myCatalogName",
      "VENDOR_NAME": "SRM Supplier",
      "VENDOR": "385",
```

```
"VENDOR_MAT": "candid 1",  
"ITEM_TYPE": "",  
"LANGUAGES": [  
  {  
    "languageCode": "EN"  
  },  
  {  
    "languageCode": "DE"  
  }  
],  
"DESCRIPTION": [  
  {  
    "languageCode": "EN",  
    "description": "Candid Power Shot A 22001"  
  },  
  {  
    "languageCode": "DE",  
    "description": "Candid Power Shot A 22001"  
  }  
],  
"SERVICE": "",  
"PRICE_VALID_FROM": "20110711080000",  
"PRICE_VALID_TO": "20120811080000",  
"PRICE": 294,  
"CURRENCY": "EUR",  
"PRICE_QUANTITY": 1,  
"PRICE_SCALES": [  
  {  
    "scaleType": "01",  
    "low": 3,
```

```
"price": 20
}
],
"TAX": 3,
"LONGTEXT": [
{
  "languageCode": "EN",
  "longtext": "An array of advanced features and functions – like the Toy Camera and Monochrome effects – gives it the flexibility to capture all your creative moods. And 14.1 megapixel resolution along with a 28mm wide-angle lens and a 4x optical zoom keep it ready for any photo opportunity. Express yourself – the PowerShot A2200"
}
],
"SEARCH_TERM": [
{
  "languageCode": "EN",
  "keyword": "Candid"
},
{
  "languageCode": "DE",
  "keyword": "Candid"
},
{
  "languageCode": "EN",
  "keyword": "Camera"
},
{
  "languageCode": "DE",
  "keyword": "Kamera"
}
] ],
"MATNR": "",
"MATGROUP": "LOC03",
```

```
"MANUFACTNAME": "Candid",
"MANUFACTCODE": "",
"MANUFACTMAT": " MANF331243421",
"LEADTIME": 2,
"UNIT": "EA",
"CONTENT_UNIT": "EA",
"PACKAGING_QTY": 1,
"MINORDER_QTY": 1,
"LOTSIZE": "",
"CONTRACT": "",
"CONTRACT_ITEM": "",
"EXT_QUOTE_ID": "",
"EXT_QUOTE_ITEM": "",
"EXT_SCHEMA_TYPE": "",
"EXT_CATEGORY_ID": "",
"EXT_CATEGORY": "",
"ATTRIBUTES": [
  {
    "categoryName": "VideoFeatures",
    "languageCode": "EN",
    "categoryValues": [
      {
        "fieldName": "Viewing Angle",
        "fieldValue": "160",
        "fieldUnit": "DD"
      }
    ]
  },
  {
    "categoryName": "AudioFeatures",
```

```
"languageCode": "EN",
"categoryValues": [
  {
    "fieldName": "Speakers",
    "fieldValue": "2 Built In",
    "fieldUnit": "EA"
  },
  {
    "fieldName": "Stereo System",
    "fieldValue": "Nicam B/G",
    "fieldUnit": "EA"
  }
]
},
"AVERAGE_RATING": 3.5,
"ASSORTMENT_IND": "01",
"PARENT_ID": "",
"ATTACHMENTS": [
  {
    "fileName": "candid-eos-1100d-slr-400x400-imad37mqyhj8jyfq.jpg",
    "mimeType": "jpg",
    "descriptions": [
      {
        "description": "FrontImagee",
        "languageCode": "EN"
      }
    ]
  },
  {
    "type": "01"
  }
],
```

```
{
  "fileName": "candid-eos-1100d-slr-400x400-imad39fk6g37fueu.jpg",
  "mimeType": "jpg",
  "descriptions": [
    {
      "description": "RearImage",
      "languageCode": "EN"
    }
  ],
  "type": "02"
},
{
  "fileName": "candid-eos-1100d-slr-400x400-imad39fkwcef5npf.jpg",
  "mimeType": "jpg",
  "descriptions": [
    {
      "description": "SideImage",
      "languageCode": "EN"
    }
  ],
  "type": "03"
}
],
"CUSTOMER_FIELDS": [
  {
    "fieldName": "xyz",
    "fieldValue": "englishOnly",
    "languageCode": "EN"
  }
],
```

```
"DELETION_IND": "",
"CHANGED_AT": "20110711080000",
"CATALOG_MANAGED": "",
"SUSTAINABLE_CMPL": "01",
"SUSTAINABLE_IND": [
  {
    "typeCode": "01",
    "value": "20",
    "unit": "CMQ"
  },
  {
    "typeCode": "02",
    "value": "30",
    "unit": "CMQ"
  }
]
}
]
```

The above example uses default page size as 20. So if there are 2002 items being served by the catalog for the timestamp, then there would be a total of 101 pages and therefore SRM Server will call the catalog 101 times with incrementing page numbers.

6.7 Additional Functions

6.7.1 Catalog Managed Items

Catalogs have the option of sending only partial data from their catalogs for mass replication. This may be required in scenarios where the price or other important attributes change too often (for example, the prices for crude oil can change every hour). In such scenarios, the catalogs can choose not to send the price data. Catalogs should send X for the parameter CATALOG_MANAGED. The item will appear in the search results but it can only be bought from the catalog provider directly. The end user will see the *Go to Catalog* button instead of the *Add to Cart* button for such items in his search results.

In order to activate this behavior, the following parameters are transferred to the product catalog on call-up:

Name	Value
FUNCTION	DETAILADD
PRODUCTID	<Database key for the product in the catalog>

Then the product catalog should immediately display the detail view of the corresponding product and on add to cart the catalog service provider has to send this information back to SAP SRM in the exact same format as specified in [Section 3.2](#). Since this field is at item level, it is possible that certain items in a single catalog are marked as CATALOG_MANAGED =X but others are replicated completely with all data. For this function to work correctly it is mandatory that the catalog service provides a web based HTTP/HTTPS interface to enable shopping of this particular item as explained in [Section 3](#).

6.7.2 Technical Specification for real-time Quantity check

When the user clicks on the Item Details/Add to Cart, the SRM system calls the catalog to determine any available stock and the expected date of delivery. The format of the URL calls is:

URL Parameter Name	Value
FUNCTION	QUANTITYCHECK
EXT_PRODUCT_ID	Catalog's unique Product ID for the item being queried
QUANTITY	Ordered Quantity. If not explicitly passed should default to 1

The catalog needs to send back the following data:

Name	Value
AVAILABLE_QUANTITY	The currently available quantity in stock
LEADTIME	Delivery time in days for the ordered quantity

An example of the HTTP response is given below:

```
{“AVAILABLE_QUANTITY”:10
“LEADTIME”=”7”}
```

6.8 Data transfer format and rules

6.8.1.1 Field Definition

The following table gives the definition of the fields that are currently supported by OCI 5.0.

Field Name	Data Type	Length	Decimals	JSON Object Structure	Description	Equivalent OCI 4.0	Mand.
EXT_PRODUCT_ID	CHAR	40	0		Unique key for this item in the catalog. This field must be provided by the catalog	NEW_ITEM-EXT_PROD UCT_ID	X

					vendor.		
CATALOG_NAME	CHAR	255	0		Catalog Name as defined by the vendor		
VENDOR_NAME	CHAR	255	0		Vendor's name, for example "ABC Pvt. Ltd."		
VENDOR	CHAR	10	0		Vendor Code	NEW_ITEM-VENDOR	
VENDOR_MAT	CHAR	40	0		Vendor Material Number	NEW_ITEM-VENDORMATERIAL	
ITEM_TYPE	CHAR	1	0		Item Type Outline, R-Root	NEW_ITEM-ITEM_TYPE	
LANGUAGES	JSON Object String	∞	0	languageCode: char2	ISO Language Codes e.g.: EN for English		X
DESCRIPTION	JSON Object String	∞	0	languageCode: char2 description: char255	ISO Language Codes Description of the Item Descriptions should be provided for all languages given in field 'LANGUAGES' or 'MATNR' should be provided. If the description is given for any language other than the ones in field 'LANGUAGES', those descriptions will be ignored.	NEW_ITEM-DESCRIPTION	

SERVICE	CHAR	1	0		Flag: X means item is a service	NEW_ITEM-SERVICE	
PRICE_VALID_FROM	DEC	15	0		UTC Time Stamp in Short Form (YYYYMMDDhhmmss). Price Validity Start Date.		
PRICE_VALID_TO	DEC	15	0		UTC Time Stamp in Short Form (YYYYMMDDhhmmss). Price Validity End Date.		
PRICE	DEC	23	4		List Price of the Item per PRICE_QUANTITY	NEW_ITEM-PRICE	
PRICE_QUANTITY	CHAR	15			Price Quantity. It is an optional field where the supplier can indicate which quantity to receive for the price. Especially with items having a very tiny price, e.g. 1 screw for 0.005 cents, you would get a price of "5,- Euro per 1000 PCE" with 1000 being the PRICE_QUANTITY. If the price quantity is not mentioned, it is assumed to be 1	NEW_ITEM-PRICEUNIT	
CURRENCY	CUKY	5	0		List price Currency sent as currency ISO code.	NEW_ITEM-CURRENCY	
PRICE_SCALES	JSON Object String	∞	0	scaleType: char2 low : char15 price: dec23	Pricing Scales based on Quantity etc. An array of scales can be sent using JSON Notation. Each such price scale contains the following Attributes: scaleType, low, high, and price. ScaleType : "01" for quantity based Scales Low : Mandatory low value		

					Price : Price		
TAX	DEC23	∞	5		Percentage tax if applicable		
LONGTEXT	JSON Object String	∞	0	languageCode:char2 longtext:string	Long description of the item	NEW_ITEM-LONGTEXT	
SEARCH_TERM	JSON Object String	∞	0	languageCode:char2 keyword:char255	JSON Object Array to specify the search terms to find this item for e.g.: [{"KeyWord": "LCD", "languageCode": "en"}, {"KeyWord": "TV", "languageCode": "EN"}]		
MATNR	CHAR	40	0		Product Number in the SRM System for this item	NEW_ITEM-MATNR	
MATGROUP	CHAR	20	0		Product Group in the SRM system for this item	NEW_ITEM-MATGROUP	
MANUFACTNAME	CHAR	255	0		External Manufacturer Name		
MANUFACTMAT	CHAR	255	0		Manufacturer Part Number	NEW_ITEM-MANUFACTMAT	
MANUFACTCODE	CHAR	10	0		Manufacturer	NEW_ITEM-MANUFACTCODE	
LEADTIME	CHAR	5	0		Delivery Time in Days	NEW_ITEM-LEADTIME	
UNIT	CHAR	3	0		Unit of Measure	NEW_ITEM-UNIT	
CONTENT_UNIT	Char	3	0		CONTENT_UNIT is normally meant for information purposes only in order to inform the end-user, that when he buys 1 Unit of a specific item, he will receive a certain amount of Content Units in the end - for example 1 Box (Unit) consists of 10 (Packaging Quantity) Pieces (Content Unit).		
PACKAGING_QTY	Char	15	0		No. of Content Unit per Order Unit		

MINORDER_QTY	CHAR	15	0		This is the minimum quantity of units that can be Ordered. Quantity less than the minimum order quantity cannot be ordered Example: If you can only order a minimum of 10 Pens then the MINORDER_QTY would be 10.		
LOTSIZE	CHAR	15	0		This indicates in which quantity increments the item can be ordered. There must be a logical compatibility between this entry and the minimum order quantity. Example: If a packet of pens contains 10 pens each and only a packet of pens can be bought. If the unit is "EA" then the lot size would be 10 and if the unit is "PAK" then the lot size would be 1.		
CONTRACT	CHAR	10	0		SRM contract to which the item refers	NEW_ITEM-CONTRACT	
CONTRACT_ITEM	CHAR	10	0		Item within the SRM contract	NEW_ITEM-CONTRACT_ITEM	
EXT_QUOTE_ID	CHAR	35	0		Number of an external bid for this item (as reference for a subsequent purchase order)	NEW_ITEM-EXT_QUOTE_ID	
EXT_QUOTE_ITEM	CHAR	10	0		Item of external bid	NEW_ITEM-EXT_QUOTE_ITEM	
EXT_SCHEMA_TYPE	CHAR	10	0		Name of a schema via which it was imported in the SRM Server	NEW_ITEM-EXT_SCHEMA_TYPE	
EXT_CATEGORY_ID	CHAR	60	0		Unique key for an external category from the schema	NEW_ITEM-EXT_CATEGORY_ID	

					above, independent of the version of the schema		
EXT_CATE GORY	CHA R	40	0		Unique key for an external category from the schema above, dependent on the version of the schema	NEW_ITEM- EXT_CATE GORY	
ATTRIBU TES	JSON Objec t String	∞	0	categoryName:char100 languageCode:char2 categoryValues :[fieldName:char100 fieldValue:char100 fieldUnit:char3	Attribute category and attributes. This will be shown on the UI. For example: "Specifications", "Accessories" Or "Dimensions". These can be used for Filtering the search in the UI . Attribute name. This is shown on the UI. For example: "Screen Size", "Weight", "Length". CategoryName: Attribute Category Name e.g Specifications languageCode : Language Code CategoryValues: An array of name value pairs in JSON Representation fieldName : Attribute Name fieldValue : Attribute Value fieldUnit : Optional ISO Code for Attribute Unit e.g: "ATTRIBUTES": [{ "categoryName": "VideoFeatures", "languageCode": "EN", "categoryValues": [{ "fieldName":		

					"AspectRatio", "fieldValue": "16:9" "fieldUnit": "mtk" }, { "fieldName": "Viewing Angle", "fieldValue": "160", "fieldUnit": "mtk" }] },":		
AVERAGE_RATING	DEC	2	1		Average Rating for this Product. For example: 2.1. Maximum possible rating is 5.0.		
ASSORTMENT_IND	Char	2	0		Assortment Indicator should be used to indicate that certain items are marked as "Preferred" or "CoreItem" The following typecodes are used 01 : Preferred		
SUSTAINABLE_COMPL	Char	2	0		Type Code Indicating the Sustainability Compliance of the item. 01 : Fully Compliant 02 : Partially Compliant 03 : Not Compliant		
SUSTAINABLE_IND	JSON Object String	∞		typeCode:char2 value:char100 unit:char03	Sustainability related fields typeCode : 01 – Direct Emission typeCode : 02 - Indirect Emission typeCode : 03 – Temporary Emission typeCode : 04 – Recycle Value typeCode : 05 – Water Consumption typeCode : 06 – Energy		

					Consumption typeCode : 07 – Waste Generated "SUSTAINABLE_ IND ": [{ "typeCode": "01", "value": "20", "unit": "DD" }, { "typeCode": "02", "value": "30", "unit": "DD" }] }		
PARENT_I D	CHA R	225	0		If this item is child item of another in a hierarchy, the parent item's EXT_PRODUCT_ ID value.	NEW_ITEM- PARENT_ID	
ATTACHM ENT_CTI	Char	02			Flag indicating in case of update if all the attachments will be sent or only the changes will be sent		
ATTACHM ENTS	JSON Objec t String	∞	0	fileName:ch ar100 mimeType:c har30 description s:[{languag eCode:char 2 description: char255}] type:char2	fileName : URL/File Name of the attachment (the attachment must be accessible for downloading under this URL or the document name If supplied as physical file) mimeType : Mime Type of the attachment descriptions:An array of descriptions in supported languages containing languageCode and description languageCode : Language code for description description :	NEW_ITEM- ATTACHME NT	

					<p>Description of the attachment</p> <p>type : 01 – Main type : 02 - Thumbnail type : 03 – Other type : 04 – Sustainability Related Certificates like Energy Star Certificate type : 05 – Other Attachments</p> <p>e.g "ATTACHMENTS ": [{"fileName":"tv_image_1.jpg","mimeType":"jpg",description ns:[{"languageCode":"en","description":"FrontImage"}, {"languageCode":"de","description":"front bild"}] ,"type":"01"},{"fileName":"tv_image_2.jpg","mimeType":"jpg",description ns:[{"languageCode":"en","description":"RearImage"}], "type":"02"}]</p>		
DELETION_IND	CHAR	1	0		The catalog provider indicates to the SRM Server that this item is no longer available in the catalog by setting the value of this flag to 'X'.		
CATALOG_MANAGED	CHAR	1	0		The catalog provider indicates to the SRM Server, that item can appear in the search results of SRM but can only be ordered from the catalog directly.		

CHANGE D_AT	DEC	21	0		Number of milli-seconds since 1st January 1970. Needed for finding out if an item which already exists in the SRM database has been updated or not. The catalog provider should update this timestamp if the item data has been updated. SRM Server stores this and compares it next time when the same item is uploaded again.		
SLD_SYS _NAME	CHA R	60			Name of a system in the System Landscape Directory (SLD)		
CUSTOM ER_FIELD S	JSON Objec t String	∞	0	fieldName:char100 fieldValue:Char100 languageCode:char2	Customer Fields specified as a JSON Array containing FieldName and FieldValue "CUSTOMER_FIELDS": [{ "fieldName": "xyz", "languageCode": "en", "fieldValue": "englishOnly" },{ "fieldName": "hazardousMaterial", "fieldValue": "X", "languageCode": "DE" }],		

Note:

1. Data type DEC represents numeric fields such as price. For example, in the specification above price field is dec. 23 with 4 decimal places precision. This means 18 digits before decimal point and 4 after decimal point. Do not use commas for thousands.
2. Data type CUKY represents the currency field sent as ISO Codes

Following is a sample of a JSON file

```
[
"NEWITEM":{
  "EXT_PRODUCT_ID": "1297767###62434_0001",
  "CATALOG_NAME": "Cameras",
  "VENDOR_NAME": "SRM Supplier",
  "VENDOR": "385",
  "VENDOR_MAT": "candid 1",
  "ITEM_TYPE": "",
  "LANGUAGES": [
    {
      "languageCode": "EN"
    },
    {
      "languageCode": "DE"
    }
  ],
  "DESCRIPTION": [
    {
      "languageCode": "EN",
      "description": "candid Power Shot A 2200"
    },
    {
      "languageCode": "DE",
      "description": "candid Power Shot A 2200"
    }
  ],
  "SERVICE": "",
  "PRICE_VALID_FROM": "20110711080000",
  "PRICE_VALID_TO": "20120811080000",
  "PRICE": 294,
```

```

"CURRENCY": "EUR",
"PRICE_QUANTITY": 1,
"PRICE_SCALES": [
  {
    "scaleType": "01",
    "low": 3,
    "high": 5,
    "price": 20
  }
],
"TAX": 3,
"LONGTEXT": [
  {
    "languageCode": "EN",
    "longtext": "An array of advanced features and functions – like the Toy Camera and Monochrome effects – gives it the flexibility to capture all your creative moods. And 14.1 megapixel resolution along with a 28mm wide-angle lens and a 4x optical zoom keep it ready for any photo opportunity. Express yourself – the PowerShot A2200"
  }
],
"SEARCH_TERM": [
  {
    "languageCode": "EN",
    "keyword": "candid"
  },
  {
    "languageCode": "DE",
    "keyword": "candid"
  },
  {
    "languageCode": "EN",
    "keyword": "Camera"
  }
]

```

```

    },
    {
      "languageCode": "DE",
      "keyword": "Kamera"
    } ],
  "MATNR": "",
  "MATGROUP": "LOC03",
  "MANUFACTNAME": "candid",
  "MANUFACTCODE": "MANF331243421",
  "MANUFACTMAT": "",
  "LEADTIME": 2,
  "UNIT": "EA",
  "CONTENT_UNIT": "EA",
  "PACKAGING_QTY": 1,
  "MINORDER_QTY": 1,
  "LOTSIZE": "",
  "CONTRACT": "",
  "CONTRACT_ITEM": "",
  "EXT_QUOTE_ID": "",
  "EXT_QUOTE_ITEM": "",
  "EXT_SCHEMA_TYPE": "",
  "EXT_CATEGORY_ID": "",
  "EXT_CATEGORY": "",
  "ATTRIBUTES": [
    {
      "categoryName": "VideoFeatures",
      "languageCode": "EN",
      "categoryValues": [
        {
          "fieldName": "Viewing Angle",

```

```
"fieldValue": "160",
  "fieldUnit": "DD"
}
],
{
  "categoryName": "AudioFeatures",
  "languageCode": "EN",
  "categoryValues": [
    {
      "fieldName": "Speakers",
      "fieldValue": "2 Built In",
      "fieldUnit": "EA"
    },
    {
      "fieldName": "Stereo System",
      "fieldValue": "Nicam B/G",
      "fieldUnit": "EA"
    }
  ]
},
"AVERAGE_RATING": 3.5,
"ASSORTMENT_IND": "01",
"PARENT_ID": "",
"ATTACHMENTS": [
  {
    "fileName": "candid-eos-1100d-slr-400x400-imad37mqyhj8jyfq.jpg",
    "mimeType": "jpg",
    "descriptions": [
```

```
{
  "description": "FrontImagee",
  "languageCode": "EN"
},
{
  "type": "01"
},
{
  "fileName": "candid-eos-1100d-slr-400x400-imad39fk6g37fueu.jpg",
  "mimeType": "jpg",
  "descriptions": [
    {
      "description": "RearImage",
      "languageCode": "EN"
    }
  ],
  "type": "02"
},
{
  "fileName": "candid-eos-1100d-slr-400x400-imad39fkwcef5npf.jpg",
  "mimeType": "jpg",
  "descriptions": [
    {
      "description": "SideImage",
      "languageCode": "EN"
    }
  ],
  "type": "03"
},
],
```

```
"CUSTOMER_FIELDS": [
  {
    "fieldName": "xyz",
    "fieldValue": "englishOnly",
    "languageCode": "EN"
  }
],
"DELETION_IND": "",
"CHANGED_AT": "20110711080000",
"CATALOG_MANAGED": "",
"SUSTAINABLE_CMPL": "01",
"SUSTAINABLE_IND": [
  {
    "typeCode": "01",
    "value": "20",
    "unit": "CMQ"
  },
  {
    "typeCode": "02",
    "value": "30",
    "unit": "CMQ"
  }
]
}
]
```

6.8.1.2 *Mandatory fields required during the Import*

The following fields are required fields to be sent from catalog in all cases:

- EXT_PRODUCT_ID: Unique Item ID on the Catalog Side
- LANGUAGES: At least one language is required and should be a valid ISO

-
- Language Code
 - Either DESCRIPTION or MATNR must be provided. If MATNR is not provided, descriptions have to be sent for all the languages sent as part of the LANGUAGES field. In case of both description and MATNR being sent the description is taken over finally into the item description. In case MATNR is not provided and description is missing for a particular language code sent as part of the LANGUAGES field, this entire item in all the languages is not considered for Indexing

6.8.1.3 Required Fields and other Validations

The following fields are required fields to be sent from catalog or can be provided by implementing BAdI (/SRMNXPCATALOG_TRANSFER_DATA). In case of any of these required fields is not present or results in a validation error the item is ignored and not indexed

- Unit is mandatory if MATNR is not provided. If the catalog is customized with Product Check switched off in the web service definition then Unit is not mandatory. In case Unit is mandatory and is provided it has to be a valid ISO code and the mapping to the corresponding SAP Commercial unit should be existing
- CURRENCY is mandatory if "CatalogManaged" is sent as false. The currency should be sent as a Valid ISO code and the corresponding conversion to SAP Code should be already maintained.
- Vendor if sent has to be already maintained as a Business Partner in SRM System
- If product check is not switched off then the MATNR sent has to exist in the product master
- If MATGROUP is provided it has to be a valid material group in the SRM System

6.8.1.4 Escaping Special Characters

If you have any special characters such as quotation marks in data, it is imperative to escape them. Otherwise the JSON parser does not work correctly. For example, if the data in a column is this example data contains a quote, then the quotation mark should be escaped with \ to signal that the enclosed data contains escaped data as given below:

"This example data contains quote".

6.8.2 Product Numbers

There are four fields in the interface that describe product numbers:

- MATNR: The product number in the SAP SRM system of the purchaser
- VENDORMAT: The supplier's product number
- MANUFACTMAT: The manufacturer's product number
- EXT_PRODUCT_ID: The number that uniquely identifies the product in the catalog.

These product numbers may not be mixed or used for other purposes, in particular the field MATNR may only be filled if the product number in the customer system is known to the catalog.

6.8.3 Configurable Products

Some products (such as laptops) can be configured in the catalog (for example at dell.com, hp.com). However, the configuration information is not part of the OCI since the structure of this information differs greatly between providers. The *Attributes* field (Nested JSON Object as specified in [Section 6.8.1.1](#)) has been added in OCI 5.0 to handle transfer of such data.

6.8.4 External Product Categories

As of SAP Enterprise Buyer 3.0, product categories from external schemas can also be used in the OCI. However, a prerequisite is the import of the relevant schema using the Product Content Workbench (PCW). If the imported schema is also used internally in the SRM Server, then no further action is required. If, however, there is to be a link from the product category from the external schema to an internal product category in the SRM Server, the mapping of the imported (or parked) schema must first be completed via the PCW. The fields EXT_SCHEMA_TYPE and EXT_CATEGORY_ID are used for the external product categories.

If, for example, an UNSPSC schema has been imported in the SRM Server under the name *UNSPSC_Vendor1*, then *UNSPSC_Vendor1* is the value that must be transferred in the field *EXT_SCHEMA_TYPE*.

6.8.5 Customer-Specific Extensions in the OCI

To transfer additional fields in the OCI:

Using the Fields *CUSTOMER_FIELDS*

These fields have no fixed semantics in the OCI and this means that you can transfer any content in order to evaluate them in BAdI */SRMNX/CATALOG_TRANSFER_DATA*.

7 Troubleshooting

When you create purchasing documents using the OCI, errors can occur that are related to the meaning or validity of the data. In order to easily identify such errors, you can activate the application log for a catalog linked using the OCI in SRM Server Customizing (for more information, see the Customizing for *Supplier Relationship Management* under *SRM Server* → *Master Data* → *Define External Web Services (Catalogs, Vendor Lists etc.)* → *Technical Settings* → *Maintain Error Log*).

You reproduce the error after you have switched on the application log. You can use transaction SLG1 to clear the application log in the SRM Server. Use *BBP_OCI* as the entry for the *Object* field. You can find more information on the application log in the F1 Help *Maintain Error Log*.

When you replicate catalog data from a physical file or via HTTP transfer (Mass Replication) all the errors during the import process are logged in the SAP Post Processing Office. The Error Log can be viewed in the Extraction Cockpit itself. In the Extraction Cockpit you can select the catalog which was replicated and get the error logs for all the replications that have happened against this catalog.