

Business Rule Framework plus (BRF Plus) XML Export and Import



Applies to:

Business Rule Framework plus (BRFplus) shipped with SAP NetWeaver 7.0 Enhancement Package 2.
For more information, visit the [Business Rules Management homepage](#).

Summary

The paper describes the XML export and import features of BRFplus. With the help of an example, the paper explains the XML export and import for standard BRFplus objects and user-defined expressions. The example covers aspects such as performing the XML import and export using the BRFplus API, backend, and the BRFplus UI integration of the functionality.

Author: Carsten Ziegler, Piyush Deora, Saurabh Kumar Pandey

Company: SAP AG

Created on: 30 July 2010

About the Authors



Carsten Ziegler is the architect and project manager of Business Rules Framework plus. He joined SAP in 2000. Since then he has been working in various projects as a developer, development architect and project lead.



Piyush Deora is a Software Engineer in the BRFplus team. He joined SAP in 2008. He is the Third Party Integration expert for BRFplus.

Table of Contents

XML Export and Import	3
Need for XML Export and Import.....	3
XML Characteristics	3
Well-Formed and Valid XML	3
Namespace and URI	4
Schema.....	4
Versioning	5
Change Tracking.....	6
Tag Hierarchy	7
Root Element:	7
Attributes:.....	7
XML Export and Import Reports	12
FDT_XML_EXPORT.....	13
Input Information.....	13
BRFplus XML Output Schema	13
FDT_XML_IMPORT	14
Input XML File.....	14
XML Import Type	14
Transport Request	15
Action Type	15
XML Info.....	15
FDT_DEBUG_XML_EXPORT_IMPORT	15
Input.....	15
Action.....	16
BRFplus XML Export and Import Using the User Interface (UI)	16
XML Export Using the UI	17
XML Import Using the UI.....	19
BRFplus APIs for XML Export and Import.....	20
EXPORT_XML.....	20
EXPORT_XML_APPLICATION	21
IMPORT_XML.....	21
BRFplus XML Export and Import Implementation	22
XML Export	22
XML Import	23
BRFplus API for XML Export or Import Implementation.....	24
EXPORT_XML.....	24
IMPORT_XML.....	24
XML Export and Import for Custom Expression Types	24
History of XML Versions	25
Related Content.....	26
Copyright	27

XML Export and Import

The BRFplus XML export and import functionality lets you transfer one or more BRFplus objects from one system to another, assuming that both systems support BRFplus. Only the current active version of the BRFplus objects can be exported. An XML file is generated for the exported objects. From this file, the objects can be imported to the target system.

Need for XML Export and Import

Although BRFplus provides transport functionality, the need for XML export and import arises in the following scenarios:

- BRFplus content needs to be transferred to other SAP systems that are not connected to the SAP transport system.
- BRFplus content needs to be published to non-SAP systems.
- BRFplus content needs to be corrected with the use of an SAP note.

XML Characteristics

Well-Formed and Valid XML

A well-formed XML document is defined as an XML document with correct XML syntax.

A valid XML is defined as an XML document that conforms to the rules of a Document Type Definition (DTD).

The XML used in BRFplus XML export and import tool is well-formed and valid. At the time of import, the XML is validated against an internal DTD. The XML elements must be defined statically (for BRFplus objects) or dynamically (for customer-defined expressions).

The DTD (Document Type Definition) for the BRFplus standard objects is shown below. The transformation can be found in development package SFDT_CORE.

Transformation FDT_APPEND_INTERNAL_DTD Active

Properties SourceCode

```

1 <xsl:transform xmlns:xsl="http://www.w3.org/1999/XSL/Transform" xmlns:sap="http://www.sap.com/sapxsl" xmlns:FDTNS="http://www.sap.com/fdt/transport"
2
3 <xsl:strip-space elements="*" />
4 <xsl:template match="@*|node()" >
5 <xsl:if test="name() = 'FDTNS:FDT'" >
6 <xsl:text disable-output-escaping="yes">
7 <![CDATA[
8 <!DOCTYPE FDTNS:FDT [
9 <!-- Additional Description for the root element FDT -->
10 <!ELEMENT FDTNS:FDT (FDTNS:COMPONENT_RELEASES*, FDTNS:APPLICATION*, FDTNS:ISSUER*, FDTNS:DATA_OBJECT*, F
11 <!-- ATTLIST FDTNS:FDT Date CDATA #IMPLIED>
12 <!-- ATTLIST FDTNS:FDT BRFplusVersion CDATA #IMPLIED>
13 <!-- ATTLIST FDTNS:FDT BRFplusInitialVersion CDATA #IMPLIED>
14 <!-- ATTLIST FDTNS:FDT BRFplusSchema CDATA #IMPLIED>
15 <!-- ATTLIST FDTNS:FDT SAPRelease CDATA #IMPLIED>
16 <!-- ATTLIST FDTNS:FDT Server CDATA #IMPLIED>
17 <!-- ATTLIST FDTNS:FDT SystemID CDATA #IMPLIED>
18 <!-- ATTLIST FDTNS:FDT Client CDATA #IMPLIED>
19 <!-- ATTLIST FDTNS:FDT SourceExportReqID CDATA #IMPLIED>
20 <!-- ATTLIST FDTNS:FDT xmlns:FDTNS CDATA #REQUIRED>
21 <!-- ATTLIST FDTNS:FDT Time CDATA #IMPLIED>
22 <!-- ATTLIST FDTNS:FDT User CDATA #IMPLIED>
23 <!-- ELEMENT FDTNS:COMPONENT_RELEASES (FDTNS:COMPONENT_RELEASE*) >
24 <!-- ELEMENT FDTNS:DATA_OBJECT (FDTNS:ADMIN_DATA, FDTNS:DATA_OBJECT_TYPE, FDTNS:DDIC_TYPENAME
25 (FDTNS:ELEMENT_TYPE, FDTNS:ALLOWED_COMPARISONS, FDTNS:OPTION
26 FDTNS:ELEMENT_LENGTH, FDTNS:ELEMENT_DECIMALS, FDTNS:ELEMENT
27 (FDTNS:STRUCTURE_ID|FDTNS:ELEMENT_ID)) >
28 <!-- ELEMENT FDTNS:EXPRESSION_TYPE (FDTNS:ADMIN_DATA, FDTNS:EXPRESSION_CLASS, FDTNS:EXPRESSION_TE
29 <!-- ELEMENT FDTNS:EXPRESSION (FDTNS:ADMIN_DATA, FDTNS:EXPRESSION_TYPE_ID, FDTNS:CONTEXT_DAT
30 ((FDTNS:OPERATION, FDTNS:PARAMETERS) | (FDTNS:OPERATION_DATA, F
31 (FDTNS:CASE_PARAMETER, FDTNS:WHEN_TABLE, FDTNS:CASE_SENSITIVI
32 | FDTNS:VALUE | (FDTNS:ENHANCEMENT_CATEGORY, FDTNS:ISSUER_ID?,
33 (FDTNS:PARAMETER_ID, FDTNS:MODE) | FDTNS:TOKENS | (FDTNS:FUNCTI
34 (FDTNS:PROBABILITY|FDTNS:PROBABILITY_ID?) |
35 (FDTNS:TEST_PARAMETER_ID, FDTNS:RANGE_TABLE, FDTNS:CASE_SENS
36 (FDTNS:FOLLOWUP_ACTIONS?, ((FDTNS:BOR?, FDTNS:CONTAINER?) | (F
37 (FDTNS:MESSAGES?, FDTNS:APPLICATION_LOG_OBJECT?, FDTNS:APPL
38 (FDTNS:CLASS?, FDTNS:SUB_EXPRESSION_IDS?) | (FDTNS:RECIPIENT
39
40 (FDTNS:TRANSFORMATION?, FDTNS:TRANSFORMATION_ID?, FDTNS:TRANSF
41 (FDTNS:CONDITION?, FDTNS:TRUE_ACTION?, FDTNS:TRUE_ACTIONS?, FDT
42 (FDTNS:TABLE_NAME, FDTNS:SINGLE, FDTNS:FIELD_CONDITIONS, FDTNS:
43 (FDTNS:FOLLOWUP_ACTIONS?, FDTNS:CHANGE_SETTINGS) |
44 (FDTNS:EXPORTING_PARAMETERS, FDTNS:CHANGING_PARAMETERS, FDTNS:
45 FDTNS:FOR_EACH_TABLE?, FDTNS:FOR_EACH_CONDITIONS?, FDTNS:RULE

```

XML Ln 1 Col 1 NUM

Namespace and URI

The namespace uniquely identifies elements in XML. BRFplus XML elements are distinguished from the customer-defined XML elements through the unique namespace and Uniform Resource Identifier (URI). BRFplus elements are defined with URI <http://sap.com/fdt/transport> and namespace **FDTNS**.

Schema

External and internal schema types are supported by BRFplus. The schema type information is stored in the root element **FDT** in the attribute **BRFplusSchema**.

- **Internal Schema Type**

This is the default schema type. The names are capitalized and an underscore is used as a word separator. Also, the values are abbreviations of the actual description. For example,

```

<FDTNS:DATA_OBJECT_TYPE>E</FDTNS:DATA_OBJECT_TYPE>
<FDTNS:ELEMENT_TYPE>T</FDTNS:ELEMENT_TYPE>

```

- **External Schema Type**

The names are formatted according to the upper camel case scheme. There is no underscore separating the words. The values are actual description of abbreviations. For example,

```
<FDTNS:DataObjectType FixedValue="E">Element</FDTNS:DataObjectType>
<FDTNS:ElementType FixedValue="T">Text</FDTNS:ElementType>
```

Versioning

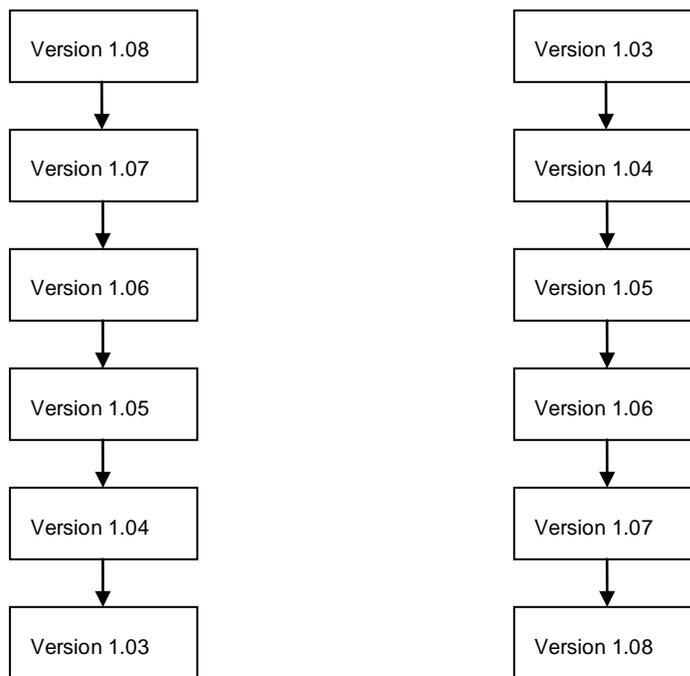
The BRFplus XML export and import tools support XML versioning. XML versioning allows version upgrading and downgrading, thus enabling BRFplus to work with SAP system that are based on different NetWeaver releases. Systems can export XML files in any version (current or lower) and import XML file of any version (current or higher or lower).

- **Upgrading**

Systems with a higher XML version can import data with a lower XML version. For example, if the current version of the source system is 1.05 and the current version of the target system is 1.07, while importing the XML file, the XML is upgraded to 1.07.

- **Downgrading**

Source systems can export content which is compatible with XML versions below the current version. For example, if the current version of the source system is 1.07, it is possible to generate lower XML versions (that is 1.06 and lower).



Version Downgrading (Can be chosen during Export)

Version Upgrading (Done automatically during Import)

The version upgrading and downgrading are done step by step.

For example: If a version needs to be downgraded from 1.08 to 1.04, then it first converts into 1.07, then to 1.06, 1.05, and finally to the required version 1.04.

Note: The BRFplus XML version governs the XML syntax which is supported in a particular BRFplus XML. The BRFplus object version indicates any change in the BRFplus object. Moreover, during the XML import, an error message is shown if the supplied XML version is greater than the XML version in the target system. The highest supported

XML version of the target system then needs to be exported from the source system. The XML version of the target system can be determined in the XML import screen of the target system.

Change Tracking

BRFplus XML export and import support the tracking of exported and imported XML files. XML Files which have transformed during the export or import and the processing messages raised during transformation are recorded to keep track of change in the files.

During the export, the generated XML is stored in the database table **FDT_XML_EXPORT** of the source system with an XML export request ID which is also written into the exported XML document.

Dictionary: Display Table

Field	Key	Initi	Data element	Data Type	Length	Decim	Short Description	Group
TRREQUEST_EXP	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	TRKORR	CHAR	20	0	Request/Task	
SOURCE_CLIENT	<input type="checkbox"/>	<input checked="" type="checkbox"/>	MANDI	CLNT	3	0	Client	
XML_VRSN	<input type="checkbox"/>	<input type="checkbox"/>	FDT_XML_VERSION	DEC	6	2	FDT: XML Version Number	
XML	<input type="checkbox"/>	<input type="checkbox"/>		STRING	0	0	String representation of Original XML	
TRANS_XML_VRSN	<input type="checkbox"/>	<input type="checkbox"/>	FDT_XML_VERSION	DEC	6	2	FDT: XML Version Number	
TRANS_XML	<input type="checkbox"/>	<input type="checkbox"/>		STRING	0	0	String representation of Transformed XML	
TRANS_MSG_XML	<input type="checkbox"/>	<input type="checkbox"/>		STRING	0	0	String representation of Transformation Message XML	

Similarly, the imported XML document is stored in the table **FDT_XML_IMPORT** of the target system. The imported XML document is stored with a generated import request ID along with the export request ID present in the document. For all objects which are changed during the XML import, the generated import request ID is also saved in the table **FDT_ADMN_0030** in field **TRREQUEST**. This completes the necessary information for being able to track the object changes from the target system back to the source system.

Dictionary: Display Table

Field	Key	Initi	Data element	Data Type	Length	Decim	Short Description	Group
TRREQUEST_IMP	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	TRKORR	CHAR	20	0	Request/Task	
TARGET_CLIENT	<input type="checkbox"/>	<input checked="" type="checkbox"/>	MANDI	CLNT	3	0	Client	
SOURCE_SYSID	<input type="checkbox"/>	<input type="checkbox"/>	SYSYSID	CHAR	8	0	Name of the SAP System	
SOURCE_CLIENT	<input type="checkbox"/>	<input type="checkbox"/>	MANDI	CLNT	3	0	Client	
TRREQUEST_EXP	<input type="checkbox"/>	<input type="checkbox"/>	TRKORR	CHAR	20	0	Request/Task	
XML_VRSN	<input type="checkbox"/>	<input type="checkbox"/>	FDT_XML_VERSION	DEC	6	2	FDT: XML Version Number	
XML	<input type="checkbox"/>	<input type="checkbox"/>		STRING	0	0	String representation of Original XML	
TRANS_XML_VRSN	<input type="checkbox"/>	<input type="checkbox"/>	FDT_XML_VERSION	DEC	6	2	FDT: XML Version Number	
TRANS_XML	<input type="checkbox"/>	<input type="checkbox"/>		STRING	0	0	String representation of Transformed XML	
MSG_XML	<input type="checkbox"/>	<input type="checkbox"/>		STRING	0	0	String representation of Import Messages in XML format	

Note: FDT_XML_EXPORT and FDT_XML_IMPORT tables are cleaned periodically with a retention period of 90 days.

Tag Hierarchy

Root Element:

The root element encloses all other elements and is the sole parent element to all other elements. The root element in BRFPplus XML is FDT.

```
<FDTNS:FDT xmlns:FDTNS="http://sap.com/fdt/transport" Client="000" Date="20080829"
BRFplusInitialVersion="1.08"
BRFplusSchema="http://sap.corp/fdt/Internal.xsd"
BRFplusVersion="1.08" SAPRelease="712" Server="server42"
SourceExportReqID="$X1P0000000000011083" SystemID="BCE" Time="053827" User="DEORA"
AP="OFF" >
```

Attributes:

The attributes of the root element are used to store the general properties of the XML file. They also provide information about the source system that generated the XML file.

Name	Description
XMLNS	The XML namespace to uniquely identify BRFPplus elements is FDTNS and the URI used is http://sap.com/fdt/transport
Client	Client ID of the source system from where the XML has been generated
Date	Date on which the XML file has been generated
BRFplusInitialVersion	Shows the current XML version of the source system from which the XML has been generated
BRFplusVersion	Shows the XML version that was requested while exporting the BRFPplus object
BRFplusSchema	Shows the schema (internal or external) of the exported file
SAPRelease	Shows the current SAP release of the source system
SourceExportReqID	XML exports have a unique export request ID which is generated in the source system and is used to store the exported XML file for documentation purposes in the database
SystemID	Shows the system ID of the source system
Time	Shows the time at which the XML file has been generated
User	Name of the user (logged into the source system and exported the data) who generated the XML file
AP	Shows whether the XML has been generated from a Business ByDesign system or not.

The root elements contain information of objects which are to be exported as child elements. The information is arranged in a sequential manner and is validated during import. The expected object sequence in the XML along with their element names are shown in the following table:

Sequence No.	BRFplus Object	XML Element Name for Internal Schema Type
1.	Application	APPLICATION
2.	Issuer	ISSUER
3.	Data Object	DATA_OBJECT
4.	Expression Type	EXPRESSION_TYPE
5.	Expression	EXPRESSION
6.	Filter	FILTER
7.	Ruleset	RULESET
8.	Function	FUNCTION
9.	Catalog	CATALOG

Note: The **ADMIN_DATA** element is common for all root child elements apart from other specific elements.

Admin Data

The admin data element is used to store common information about each individual BRFplus objects which is exported.

```

<FDTNS:ADMIN_DATA>
  <FDTNS:APPLICATION_ID>0013212003FC02DB8AA7CB7F4D619AC2
  </FDTNS:APPLICATION_ID>
  <FDTNS:ID>0013212003FC02DB8AB782B628BD0157</FDTNS:ID>
  <FDTNS:NAME>EMPLOYEE_ID</FDTNS:NAME>
  <FDTNS:SYSTEM_OBJECT>X</FDTNS:SYSTEM_OBJECT>
  <FDTNS:CREATION_USER>ZIEGLERCA</FDTNS:CREATION_USER>
  <FDTNS:CREATION_TIMESTAMP>20070929112908</FDTNS:CREATION_TIMESTAMP>
  <FDTNS:CHANGE_USER>ZIEGLERCA</FDTNS:CHANGE_USER>
  <FDTNS:CHANGE_TIMESTAMP>20080717194229</FDTNS:CHANGE_TIMESTAMP>
  <FDTNS:VERSIONS>
    <FDTNS:item>
      <FDTNS:VERSION>000004</FDTNS:VERSION>
      <FDTNS:USER>ZIEGLERCA</FDTNS:USER>
      <FDTNS:TIMESTAMP>20080717194227</FDTNS:TIMESTAMP>
      <FDTNS:STATE>A</FDTNS:STATE>
      <FDTNS:TRREQUEST />
      <FDTNS:TRVERSION>000004</FDTNS:TRVERSION>
      <FDTNS:TRTIMESTAMP>20080717194227</FDTNS:TRTIMESTAMP>
      <FDTNS:TRSYSID>BCE</FDTNS:TRSYSID>
      <FDTNS:TRCLIENT />
      <FDTNS:OVERS_ID>001CC412326A02EC83EEACAF62D48002</FDTNS:OVERS_ID>
      <FDTNS:CUSTOMER_CHANGE />
      <FDTNS:VERSIONING />
    </FDTNS:item>
  </FDTNS:VERSIONS>
  <FDTNS:ACCESS_LEVEL>APPL</FDTNS:ACCESS_LEVEL>
  <FDTNS:LOCAL />
  <FDTNS:TEXT_DEPENDENCY_TYPE>1</FDTNS:TEXT_DEPENDENCY_TYPE>
  <FDTNS:SHORT_TEXTS>
    <FDTNS:item>
      <FDTNS:LANGU />
      <FDTNS:TEXT>Empl. ID</FDTNS:TEXT>
    </FDTNS:item>
  </FDTNS:SHORT_TEXTS>
  <FDTNS:TEXTS>
    <FDTNS:item>
      <FDTNS:LANGU />
      <FDTNS:TEXT>Employee ID</FDTNS:TEXT>
    </FDTNS:item>
  </FDTNS:TEXTS>
  <FDTNS:DOCUMENTATION_DEPENDENCY_TYPE>1
  </FDTNS:DOCUMENTATION_DEPENDENCY_TYPE>
  <FDTNS:DOCUMENTATIONS />
</FDTNS:ADMIN_DATA>

```

The following sub-elements are used within the admin data element:

Name	Description
APPLICATION_ID	Stores the application ID to which the object belongs
ID	Stores the ID of the exported object
NAME	Stores the name of the object being exported
SYSTEM_OBJECT	Indicates whether the object is a system object or not. If the object is a system object then the value of this element is 'X' (TRUE).
MASTERDATA_OBJECT	Indicates whether the object is a master data object or not. If the object is a master data object then the value of this element is 'X' (TRUE).
CREATION_TIMESTAMP	Stores the timestamp at which the object has been created
CREATION_USER	Stores the name of the user who created the object which is being exported
CHANGE_TIMESTAMP	Stores the time at which the object has last been changed
VERSIONS	Stores the version details of the exported object
ACCESS_LEVEL	Stores the access level of the exported object
LOCAL	Indicates whether the exported object is a local object or not
TEXT_DEPENDENCY_TYPE	Stores the dependency type of the texts of an object. The dependency can be: <ul style="list-style-type: none"> • Independent of language and version • Language-dependent but not version-dependent • Version-dependent but not language-dependent • Language-dependent and version-dependent
TEXTS	Stores the long texts associated with an object. This element contains 0 to n ITEM elements which contain the actual text string for a particular

	combination of language and version, depending on the TEXT_DEPENDENCY_TYPE setting.
SHORT_TEXTS	Stores the short text associated with an object. This element contains 0 to n ITEM elements which contain the actual short text string for a particular combination of language and version, depending on the TEXT_DEPENDENCY_TYPE setting.
DOCUMENTATION_DEPENDENCY_TYPE	Stores the dependency type of the documentation of an object The dependency can be: <ul style="list-style-type: none"> • Independent of language and version • Language-dependent but not version-dependent • Version-dependent but not language-dependent • Language-dependent and version-dependent
DOCUMENTATIONS	Stores the documentation associated with the object. This element contains 0 to n ITEM elements which contain the documentation for a particular combination of language and version, depending on the DOCUMENTATION_DEPENDENCY_TYPE setting.

XML Export and Import Reports

XML export and import reports provide an easy and user-friendly way to export and import BRFplus objects. The following reports are available for XML export and import:

Report Name	Description
FDT_XML_EXPORT	Exports BRFplus content to an XML file.
FDT_XML_IMPORT	Imports information about the BRFplus object content from an XML file document into the target system.
FDT_DEBUG_XML_EXPORT_IMPORT	Debugs and tracks changes done by the XML import.

FDT_XML_EXPORT

Program to Export BRFplus XML

UI for BRFplus XML Export

Input Information

BRFplus Object ID to be exported

Read Objects from Transport Request

Export Subordinate Objects

Export Surrounding Application

XML Version No

BRFplus XML Output Schema

Internal

External

Output Type

Download XML

Display XML

This report runs in the source system where the BRFplus objects to be exported reside. For exporting the BRFplus objects, there are several fields which require the user's input.

Input Information

The object's ID has to be entered in the *BRFplus Object ID to be exported* field. An option is also provided to enter multiple IDs by using the arrow button on the right.

The ID can also be directly selected from the *Read Objects from Transport Request* field.

If the object has dependencies then the *Export Subordinate Objects* checkbox can be checked to include the subordinate objects for export.

If the *Export Surrounding Application* checkbox is marked the application of the object will be exported..

By choosing an XML Version different from the default version an XML file can be generated for downgrading the content. By default, the current version of the system is shown.

BRFplus XML Output Schema

An option to choose the schema of the XML file is available. The schema can either be external or internal.

Output Type

The output type provides two options for the XML file which is generated from the export.

- Download the XML file
- Display the XML file

FDT_XML_IMPORT

Program to Import BRFPplus XML

UI for BRFPplus XML Import

Input XML File

XML File Path*

*Note - If not provided a popup appears during execution

XML Import Type

Import Type

Transport Request

Workbench Transport Request*

*Note - Mandatory if importing xml contains system objects

Customizing Transport Request*

*Note - Mandatory if importing xml contains customizing objects

Action Type

Check Before Import

Save & Activate after Import

XML Info

System XML Version Number 1.08

This report is used to import the XML file in the target system. There are several sections or fields that require the user's input.

Input XML File

The path of the XML file to be imported has to be entered in the *XML File Path* field. To easily locate the file on your frontend server, use the value help option.

XML Import Type

This section or field describes the import type. There are three options available to import the XML file:

- **Standard**
With this option you can import changes for objects that are changeable in the target system.
- **Repair**
The repair option can be used to repair content that is normally not changeable in the system due to the fact that it was imported to the system by a normal transport. The repair option should only be used to import content changes provided by an SAP correction note. The functionality is used in order to be able to apply a content correction to a system before the corresponding support package with the corrected content is imported to the system.
- **Local Copy**
This option will create a local copy of a non-local application in the source file. A precondition for the import is that the application in the source file is not local and did not reach the target system via a normal transport. This check ensures that local copies will not overwrite transported data.

Transport Request

The imported objects need to be written to a transport request in case they were not local in the source system. If you create a local copy you do not need to specify a transport request.

A workbench transport request is used for system objects and a customizing transport request for customizing objects.

Action Type

The consistency of the XML file can be checked before the file contents is saved and activated.

- Check Before Import

The "Check Before Import" option can be used to check file consistency and the imported objects. The objects are not saved or activated in the target system.

- Save & Activate after import

The "Save & Activate after import" option can be used to import and activate the objects in the target system. Check for the file consistency and imported objects will be performed before activating the objects.

XML Info

This section shows the current XML version of the system. This information is needed if a downgrade needs to be executed at the source system side during the export of the content.

FDT_DEBUG_XML_EXPORT_IMPORT

The screenshot shows the SAP program interface for 'Program for Debugging BRFplus XML Export/Import'. The menu bar includes 'Program', 'Edit', 'Goto', 'System', and 'Help'. The toolbar contains various icons for file operations and debugging. The main area is titled 'UI for Debugging FDT XML Export/Import' and is divided into 'Input' and 'Action' sections.

Input	
XML Export/Import Transport Req No	<input type="text"/>

Action	
Display XML	<input checked="" type="radio"/>
Download XML	<input type="radio"/>
Display Transformed XML	<input type="radio"/>
Download Transformed XML	<input type="radio"/>
Display Message XML	<input type="radio"/>
Download Message XML	<input type="radio"/>

This report runs in the system which supplies the BRFplus export or import transport request number for analyzing.

Input

The transport request number for debugging is entered in the *XML Export/Import Transport Req No* field.

Action

There are six options available for the debugging process.

- Display XML
- Download XML
- Display Transformed XML
- Download Transformed XML
- Display Message XML
- Download Message XML

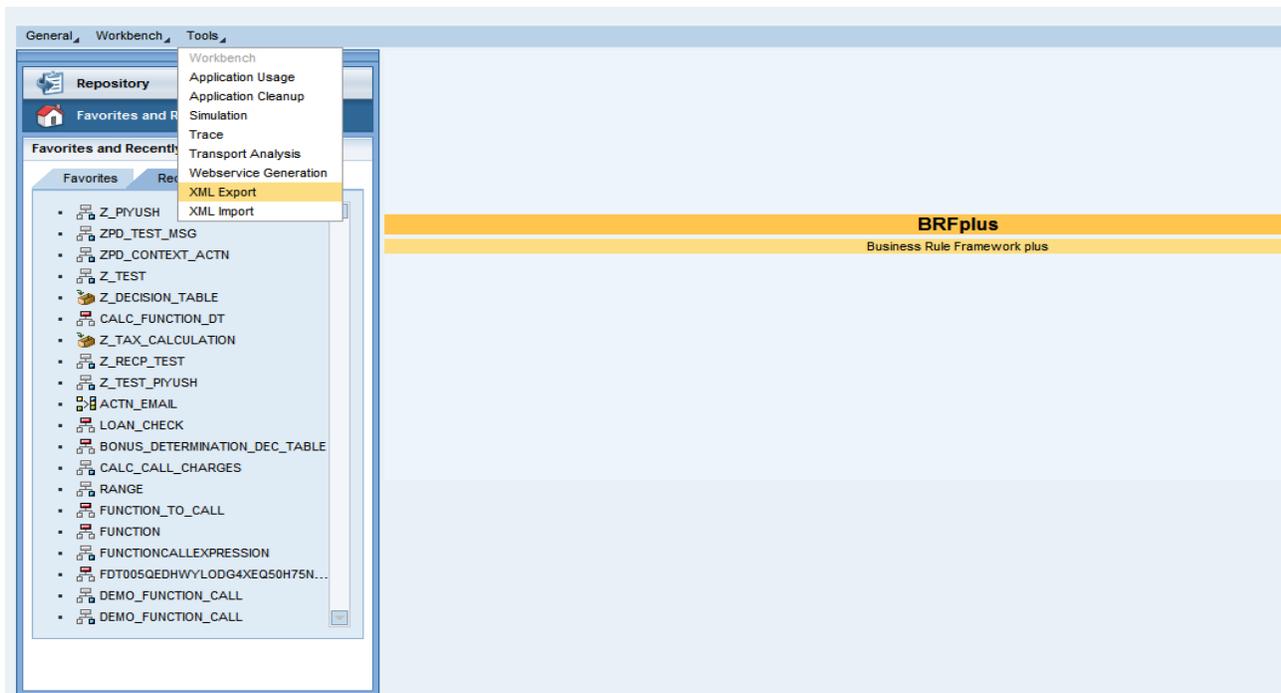
Transformed XML contains the contents after the transformation is performed on the XML file. XML downgrading is performed if XML is exported for a lower XML version and XML upgrading is performed if an XML of a lower version than the current XML version is imported into the target system.

Message XML contains the processing messages raised during XML export or Import.

BRFplus XML Export and Import Using the User Interface (UI)

Using the BRFplus workbench, BRFplus objects can be exported and imported. Transaction **FDT_WORKBENCH** instantiates the BRFplus workbench.

Note: XML Export and XML Import options are only visible in Expert Mode.



XML Export Using the UI

1. In the *Tools* menu, choose *XML Export*.
The *XML Export* page opens.

2. In the *XML File Export* section, choose *Select an object* button.
3. In the *Object Query* dialog box that appears, enter the name of the object you want to export and choose *Search*.

The results are shown in a tabular format.

Name	Status	Type	Short Text	Application
▶ ELEMENT_ID	■	Data Object		ZPD_TEST
▶ STRUCTURE_ID	■	Data Object		ZPD_TEST
▶ TABLE_ID	■	Data Object		ZPD_TEST
▶ ZPD_AMOUNT	■	Data Object		ZPD_TEST
▶ ZPD_AMT	■	Data Object		ZPD_TEST
▶ ZPD_BOOL	■	Data Object		ZPD_TEST
▪ ZPD_DECISION_TABLE	■	Expression		ZPD_TEST
▪ ZPD_FORMULA	■	Expression		ZPD_TEST
▪ ZPD_FUNCTION	■	Function		ZPD_TEST
▶ ZPD_NUMBER	■	Data Object		ZPD_TEST

4. Select the object from the table and choose *Select*.

The selected object for which the XML file will be generated appears in the *Object* field. Check the *Include child objects*, if appropriate, and select the version and schema.

5. Choose *Generate XML File*.

The system displays a file download dialog box. Choose *Save* and enter a name for the XML file.

Tools Help

XML Export

Step 1. Select an Object

Object: ZPD_FUNCTION (Test Fuction)

Include Referenced Objects:

Include Surrounding Application:

XML Version: 1.08

XML Schema: Internal

Step 2. Generate XML File

Step 3. Download XML File

```

<?xml version="1.0" encoding="utf-8" ?>
- <FDTNS:FDT Date="20090519" Time="105407" User="DEORA" Server="vmw2782" BRFplusVersion="1.08" BRFplusInitialVersion="1.08"
  BRFplusSchema="http://sap.corp/fdt/Internal.xsd" SAPRelease="720" SystemID="Y2Z" Client="000" AP="OFF" xmlns:FDTNS="http://sap.com/fdt/transport"
  SourceExportReqID="$X1P0000000000001237">
- <FDTNS:COMPONENT_RELEASES>
- <FDTNS:COMPONENT_RELEASE>
  <FDTNS:COMPONENT>SAP_BASIS</FDTNS:COMPONENT>
  <FDTNS:RELEASE>720</FDTNS:RELEASE>
  <FDTNS:VERSION>0000</FDTNS:VERSION>
</FDTNS:COMPONENT_RELEASE>
</FDTNS:COMPONENT_RELEASES>
- <FDTNS:DATA_OBJECT>
- <FDTNS:ADMIN_DATA>
  <FDTNS:APPLICATION_ID>0050568311721DEE8CF9A51435535191</FDTNS:APPLICATION_ID>
  <FDTNS:ID>0050568311721DEE8D952D09F5E71191</FDTNS:ID>
  <FDTNS:NAME />
  <FDTNS:MASTERDATA_OBJECT>X</FDTNS:MASTERDATA_OBJECT>
  <FDTNS:CREATION_USER>DEORA</FDTNS:CREATION_USER>
  <FDTNS:CREATION_TIMESTAMP>20090429103522</FDTNS:CREATION_TIMESTAMP>
  <FDTNS:CHANGE_USER>DEORA</FDTNS:CHANGE_USER>
  <FDTNS:CHANGE_TIMESTAMP>20090429103529</FDTNS:CHANGE_TIMESTAMP>
- <FDTNS:VERSIONS>
- <FDTNS:item>
  <FDTNS:VERSION>000001</FDTNS:VERSION>
  <FDTNS:USER>DEORA</FDTNS:USER>
  <FDTNS:TIMESTAMP>20090429103529</FDTNS:TIMESTAMP>
  <FDTNS:STATE>A</FDTNS:STATE>
  <FDTNS:TRREQUEST />
  <FDTNS:TRVERSION>000001</FDTNS:TRVERSION>
  <FDTNS:TRTIMESTAMP>20090429103529</FDTNS:TRTIMESTAMP>
  <FDTNS:TRSYSID>Y2Z</FDTNS:TRSYSID>
  <FDTNS:TRCLIENT>000</FDTNS:TRCLIENT>
  <FDTNS:OVERS_ID />
  <FDTNS:CUSTOMER_CHANGE />
  <FDTNS:VERSIONING />
</FDTNS:item>
</FDTNS:VERSIONS>
  <FDTNS:ACCESS_LEVEL>APPL</FDTNS:ACCESS_LEVEL>
  <FDTNS:LOCAL>X</FDTNS:LOCAL>
  <FDTNS:TEXT_DEPENDENCY_TYPE>1</FDTNS:TEXT_DEPENDENCY_TYPE>
- <FDTNS:TEXTS>
- <FDTNS:item>
  <FDTNS:LANGU />
  <FDTNS:TEXT>DEFAULT_RESULT_DATA_OBJECT</FDTNS:TEXT>
</FDTNS:item>
</FDTNS:TEXTS>
  <FDTNS:SHORT_TEXTS />
  <FDTNS:DOCUMENTATION_DEPENDENCY_TYPE>0</FDTNS:DOCUMENTATION_DEPENDENCY_TYPE>
  <FDTNS:DOCUMENTATIONS />
</FDTNS:ADMIN_DATA>
  <FDTNS:DATA_OBJECT_TYPE>S</FDTNS:DATA_OBJECT_TYPE>
- <FDTNS:ELEMENTS>

```

XML Import Using the UI

1. In the *Tool* menu, choose *XML Import*.

The *XML Import* page is displayed.

1. Under the *XML File Import* section, choose *Browse* and select the path of the XML file you want to import.
2. Depending on the objects in the imported file, provide a *Customizing Transport Request* (if the object is a customizing object) or a *Workbench Transport Request* (if the object is a system object).
3. Check the *Test Run* checkbox.

You can mark the checkbox if you want to perform only a simulation of the import without saving and activating the objects to the system.

4. Choose *Upload file*.

On successful import of the objects, a success message is displayed along with an auto-generated transport request ID.

The XML file is saved against the transport request ID in the database table **FDT_XML_IMPORT**.

In case of a failure, an error message is displayed with the information of the failed objects and the reason for the failure.

BRFplus APIs for XML Export and Import

There are several sets of interfaces, procedures and classes provided to let you take advantage of BRFplus XML export and import.

Interface: IF_FDT_DATA_EXCHANGE

The BRFplus API involved in the export and import is **IF_FDT_DATA_EXCHANGE**. The interface comprises the following methods.

Method	Description
EXPORT_XML	Exports BRFplus contents to an XML document
EXPORT_XML_APPLICATION	Exports XML for an application including all contained objects
IMPORT_XML	Imports an XML document.

EXPORT_XML

The input parameters for this method are:

Name	Description
ITS_OBJECT_ID	Provides a table of the IDs of the objects which are to be exported
IV_DEEP (Optional)	Indicates whether subordinate named objects should be exported or not
IV_SCHEMA (Optional)	Specifies the XML schema type expected as output It has two possible values: GC_XML_SCHEMA_TYPE_INTERNAL and GC_XML_SCHEMA_TYPE_EXTERNAL . GC_XML_SCHEMA_TYPE_INTERNAL is the default value.
IV_TIMESTAMP	Timestamp at which the BRFplus object's active version is selected. If the timestamp is not provided, the latest active version is selected.
IV_XML_VERSION (Optional)	Used for downgrading. Specifies the XML version which is to be exported. Do not confuse the XML version with the BRFplus object version. The XML version only governs the XML syntax in the exported BRFplus XML document. By default, the highest XML version is exported.

ITS_OBJECT_ISSUER	Provides a table of the pairs Objects IDs and Issuer IDs which are to be exported.
--------------------------	--

The output parameters are as follows:

Name	Description
EO_DOM_TREE	XML document in DOM object tree representation.
EV_STRING	XML document in string representation.
ET_MESSAGE	Processing messages for the export.
ETS_FAILURE	Provides a list of object IDs for which the export has failed

EXPORT_XML_APPLICATION

The input and output parameters are the same as **EXPORT_XML**. The method **EXPORT_XML_APPLICATION** is used when all the assigned objects of an application need to be exported with the application.

IMPORT_XML

The input parameters for this method are:

Name	Description
IO_DOM_TREE (Optional)	XML document used for the import – represented as a DOM object tree
IV_STRING (Optional)	XML document used for the import represented as an XML string
IV_CREATE (Optional)	Indicates whether BRFplus objects are created if they are not present By default, it is set as abap_true .
IV_ACTIVATE (Optional)	Indicates whether BRFplus objects need are activated after the import. The default is set as abap_true .
IV_SIMULATE (Optional)	Indicates whether the import is a simulation or not The default is set as abap_false

The first two and the last three of these import parameters are mutually exclusive. At the same time, from both parameter groups, there must be exactly one parameter provided.

The output parameters are:

Name	Description
ETS_SUCCESS	Provides a sorted list of BRFplus object IDs which are imported successfully
ETS_FAILURE	Provides a sorted list of BRFplus object IDs which are not imported successfully

The exception **CX_FDT_INPUT** is triggered if the import did not take place due to parse errors or for any other reason. The attribute **MT_MESSAGE** will then contain additional information on the cause of the problem.

BRFplus XML Export and Import Implementation

The following section explains the steps performed internally by BRFplus during XML export and import for the list of object IDs.

- XML Export
- XML Import

XML Export

The following steps are performed internally by BRFplus when the XML file is exported.

1. Checks the XML version number

Checks whether the BRFplus XML version requested is correct and supported

2. Adds root elements and its attributes

Adds the BRFplus root element, namespace declaration and root attributes

3. Adds information on component releases

Adds information about the installed components in the source system

4. Creates list of BRFplus object IDs in a particular sequence

5. Performs mass check on the object list created

6. Writes individual object content in the XML document

Calls the method **EXPORT_XML** of **IF_FDT_DATA_EXCHANGE**

7. XML transformation to the requested XML version

Performs a transformation if the XML version requested is different from the latest XML version (default) in the source system

8. XML transformation from internal schema type to external schema type (if required)

Performs a transformation if the output schema type requested is not internal (default)

9. Saves the exported XML document against the XML export transport request number

XML Import

The following steps are performed internally by BRFplus when the XML file is imported.

1. **Checks the XML format and BRFplus XML version**
2. **XML transformation from external schema type to internal schema type (if required)**

The XML import logic works according to the internal schema type and the latest BRFplus XML version in the target system. Whenever an import happens, the schema type is checked to verify whether the schema type is internal or not. The XML version checks the current version of the BRFplus XML. If the above conditions are not satisfied then a transformation is done. If any errors come up during the transformation then the import is stopped and an error message is shown.

3. **XML transformation of imported XML document to current BRFplus XML version (if required)**
4. **Validates the XML document against DTD.**

Checks if the XML document syntax is consistent

The document is checked against an internal DTD. The content of this DTD comes from the following sources:

- Pre-defined BRFplus DTD elements, that is **XSLT_FDT_APPEND_INTERNAL_DTD**
- Method **GET_DTD** of the class that implements the interface **IF_FDT_DATA_EXCHANGE_EXTERNAL** for user-defined expression type

5. **Imports BRFplus objects from the supplied XML document**
6. **Performs mass check on the imported objects**
7. **Performs mass activation on the imported objects (if requested)**
8. **Performs mass save on the imported objects (if requested)**
9. **Saves the imported XML document against the XML export transport request number**

BRFplus API for XML Export or Import Implementation

Interface: IF_FDT_DATA_EXCHANGE_INTERNAL

This interface needs to be implemented by all BRFplus object types including user-defined expression and action types that need to take part in XML export and import. The interface involves the following methods.

Method	Description
EXPORT_XML	Exports information about the BRFplus objects to the XML document
IMPORT_XML	Imports information about the BRFplus objects from the XML document

EXPORT_XML

The input parameters for this method are:

Name	Description
IV_TIMESTAMP	Specifies the active timestamp version
IO_PARENT	Describes the XML element instance in which the content needs to be appended
IO_NAMESPACE_ATTR (OPTIONAL)	Sets the user-defined namespace and URI by referring to the namespace attribute of the root element This parameter is used only in user-defined expression types.

The exception **CX_FDT_INPUT** is thrown when the export does not take place for a particular ID.

IMPORT_XML

The input parameter for this method is **IO_PARENT**. This parameter is the XML element instance from which the content needs to be read and imported.

The exception **CX_FDT_INPUT** is thrown if the import does not take place due to parse errors or for some other reasons.

XML Export and Import for Custom Expression Types

A custom expression type is an expression type which is defined by the user. The custom expression type does not have any statically-defined attributes in BRFplus that need to appear in the XML file for XML export and import.

While creating the implementation class for the custom expression type, the user needs to inherit the class **CL_FDT_EXPRESSION** or implement the interface **IF_FDT_EXPRESSION**. If it inherits from the class **CL_FDT_EXPRESSION** then the interface **IF_FDT_DATA_EXCHANGE_INTERNAL** is present. Otherwise, the interface has to be implemented separately.

The code sample for creating user-defined expression types is available in the *How to Create Custom Expression Types* document.

History of XML Versions

BRFplus XML Version No	Changes
1.01	Initial Version
1.02	Addition of Development Package for Application
1.03	Addition of short texts Addition of text dependency type Addition of document dependency type
1.04	Addition of local objects
1.05	Change in position of local objects Addition of text symbol Addition of SAPSCRIPT object Addition of versioning switch
1.06	Change in position of local objects Addition of versioning switch
1.07	Change in position of local objects. Addition of versioning switch.
1.08	Change in position of local objects. Change in Attribute FormulaDerivationTool to BRFplus.

Related Content

- [BRFplus – The Very Basics](#)
- Carsten Ziegler, *About Business Rules*:
<https://www.sdn.sap.com/irj/sdn/weblogs?blog=/pub/wlg/9713>
- Carsten Ziegler, *BRFplus a Business Rule Engine written in ABAP*,
<https://www.sdn.sap.com/irj/sdn/weblogs?blog=/pub/wlg/8889>
- Carsten Ziegler, Important Information for Using BRFplus
<https://www.sdn.sap.com/irj/sdn/weblogs?blog=/pub/wlg/11632>
- Online Help:
http://docudelivery.wdf.sap.corp:1080/saphelp_docs/Netweaver/nw70_Ehp01/index/en/helpdata/en/c/c/85414842c8470bb19b53038c4b5259/frameset.htm
- For more information, visit the [Business Rules Management homepage](#).

Copyright

© Copyright 2010 SAP AG. All rights reserved.

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

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

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

IBM, DB2, DB2 Universal Database, System i, System i5, System p, System p5, System x, System z, System z10, System z9, z10, z9, iSeries, pSeries, xSeries, zSeries, eServer, z/VM, z/OS, i5/OS, S/390, OS/390, OS/400, AS/400, S/390 Parallel Enterprise Server, PowerVM, Power Architecture, POWER6+, POWER6, POWER5+, POWER5, POWER, OpenPower, PowerPC, BatchPipes, BladeCenter, System Storage, GPFS, HACMP, RETAIN, DB2 Connect, RACF, Redbooks, OS/2, Parallel Sysplex, MVS/ESA, AIX, Intelligent Miner, WebSphere, Netfinity, Tivoli and Informix are trademarks or registered trademarks of IBM Corporation.

Linux is the registered trademark of Linus Torvalds in the U.S. and other countries.

Adobe, the Adobe logo, Acrobat, PostScript, and Reader are either trademarks or registered trademarks of Adobe Systems Incorporated in the United States and/or other countries.

Oracle is a registered trademark of Oracle Corporation.

UNIX, X/Open, OSF/1, and Motif are registered trademarks of the Open Group.

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

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

Java is a registered trademark of Sun Microsystems, Inc.

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

SAP, R/3, SAP NetWeaver, Duet, PartnerEdge, ByDesign, SAP Business ByDesign, 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.

Business Objects and the Business Objects logo, BusinessObjects, Crystal Reports, Crystal Decisions, Web Intelligence, Xcelsius, and other Business Objects products and services mentioned herein as well as their respective logos are trademarks or registered trademarks of Business Objects S.A. in the United States and in other countries. Business Objects is an SAP company.

All other product and service names mentioned are the trademarks of their respective companies. Data contained in this document serves informational purposes only. National product specifications may vary.

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