BRFplus 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.

Authors: Carsten Ziegler, Piyush Deora, Saurabh Kumar Pandey
Company: SAP AG
Created on: 22 June 2011

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 ...............................................................................................................................4
  Change Tracking .....................................................................................................................5

Tag Hierarchy ...........................................................................................................................6
  Root Element..........................................................................................................................6
  Attributes ...............................................................................................................................6

Admin Data ..............................................................................................................................8

XML Export and Import Reports..............................................................................................10
  FDT_XML_EXPORT ................................................................................................................11
    Input Information................................................................................................................11
    BRFplus XML Output Schema .........................................................................................11
  FDT_XML_IMPORT ................................................................................................................12
    Input XML File ..................................................................................................................12
    XML Import Type ...............................................................................................................12
    Transport Request .............................................................................................................12
    Action Type ........................................................................................................................13
    XML Info ............................................................................................................................13
  FDT_DEBUG_XML_EXPORT_IMPORT ....................................................................................13
    Input ....................................................................................................................................13
    Action ................................................................................................................................13

BRFplus XML Export and Import Using the User Interface (UI)............................................14
  XML Export Using the UI .....................................................................................................14
  XML Import Using the UI .....................................................................................................16

BRFplus APIs for XML Export and Import ...........................................................................17
  EXPORT_XML ......................................................................................................................17
  EXPORT_XML_APPLICATION ..............................................................................................18
  IMPORT_XML .......................................................................................................................18

BRFplus XML Export and Import Implementation ..................................................................19
  XML Export ..........................................................................................................................19
  XML Import ..........................................................................................................................20

BRFplus API for XML Export or Import Implementation ......................................................20
  EXPORT_XML ......................................................................................................................21
  IMPORT_XML .......................................................................................................................21

XML Export and Import for Custom Expression Types .........................................................21

History of XML Versions .......................................................................................................21

Related Content .....................................................................................................................23

Copyright ................................................................................................................................24
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.

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<xsd:element name="FST" type="bpm:FSTType"/>
<xsd:element name="FSTType" type="bpm:FSTTypeType"/>
<xsd:element name="FSTDataObject" type="bpm:FSTDataObjectType"/>
<xsd:element name="FSTExpression" type="bpm:FSTExpressionType"/>
<xsd:element name="FSTExpressionObject" type="bpm:FSTExpressionObjectType"/>
<xsd:element name="FSTParameter" type="bpm:FSTParameterType"/>
<xsd:element name="FSTParameterObject" type="bpm:FSTParameterObjectType"/>
<xsd:element name="FSTCondition" type="bpm:FSTConditionType"/>
<xsd:element name="FSTConditionObject" type="bpm:FSTConditionObjectType"/>
<xsd:element name="FSTCase" type="bpm:FSTCaseType"/>
<xsd:element name="FSTCaseObject" type="bpm:FSTCaseObjectType"/>
<xsd:element name="FSTCaseParameter" type="bpm:FSTCaseParameterType"/>
<xsd:element name="FSTCaseParameterObject" type="bpm:FSTCaseParameterObjectType"/>
<xsd:element name="FSTCaseCondition" type="bpm:FSTCaseConditionType"/>
<xsd:element name="FSTCaseConditionObject" type="bpm:FSTCaseConditionObjectType"/>
<xsd:element name="FSTCaseConditionObject" type="bpm:FSTCaseConditionObjectType"/>
<xsd:element name="FSTCaseConditionObject" type="bpm:FSTCaseConditionObjectType"/>
<xsd:element name="FSTCaseConditionObject" type="bpm:FSTCaseConditionObjectType"/>
<xsd:element name="FSTCaseConditionObject" type="bpm:FSTCaseConditionObjectType"/>
<xsd:element name="FSTCaseConditionObject" type="bpm:FSTCaseConditionObjectType"/>
<xsd:element name="FSTCaseConditionObject" type="bpm:FSTCaseConditionObjectType"/>
<xsd:element name="FSTCaseConditionObject" type="bpm:FSTCaseConditionObjectType"/>
<xsd:element name="FSTCaseConditionObject" type="bpm:FSTCaseConditionObjectType"/>
<xsd:element name="FSTCaseConditionObject" type="bpm:FSTCaseConditionObjectType"/>
<xsd:element name="FSTCaseConditionObject" type="bpm:FSTCaseConditionObjectType"/>
<xsd:element name="FSTCaseConditionObject" type="bpm:FSTCaseConditionObjectType"/>
<xsd:element name="FSTCaseConditionObject" type="bpm:FSTCaseConditionObjectType"/>
<xsd:element name="FSTCaseConditionObject" type="bpm:FSTCaseConditionObjectType"/>
<xsd:element name="FSTCaseConditionObject" type="bpm:FSTCaseConditionObjectType"/>
<xsd:element name="FSTCaseConditionObject" type="bpm:FSTCaseConditionObjectType"/>
<xsd:element name="FSTCaseConditionObject" type="bpm:FSTCaseConditionObjectType"/>
<xsd:element name="FSTCaseConditionObject" type="bpm:FSTCaseConditionObjectType"/>
<xsd:element name="FSTCaseConditionObject" type="bpm:FSTCaseConditionObjectType"/>
<xsd:element name="FSTCaseConditionObject" type="bpm:FSTCaseConditionObjectType"/>
<xsd:element name="FSTCaseConditionObject" type="bpm:FSTCaseConditionObjectType"/>
<xsd:element name="FSTCaseConditionObject" type="bpm:FSTCaseConditionObjectType"/>
<xsd:element name="FSTCaseConditionObject" type="bpm:FSTCaseConditionObjectType"/>
<xsd:element name="FSTCaseConditionObject" type="bpm:FSTCaseConditionObjectType"/>
<xsd:element name="FSTCaseConditionObject" type="bpm:FSTCaseConditionObjectType"/>
<xsd:element name="FSTCaseConditionObject" type="bpm:FSTCaseConditionObjectType"/>
<xsd:element name="FSTCaseConditionObject" type="bpm:FSTCaseConditionObjectType"/>
<xsd:element name="FSTCaseConditionObject" type="bpm:FSTCaseConditionObjectType"/>
<xsd:element name="FSTCaseConditionObject" type="bpm:FSTCaseConditionObjectType"/>
<xsd:element name="FSTCaseConditionObject" type="bpm:FSTCaseConditionObjectType"/>
<xsd:element name="FSTCaseConditionObject" type="bpm:FSTCaseConditionObjectType"/>
<xsd:element name="FSTCaseConditionObject" type="bpm:FSTCaseConditionObjectType"/>
<xsd:element name="FSTCaseConditionObject" type="bpm:FSTCaseConditionObjectType"/>
<xsd:element name="FSTCaseConditionObject" type="bpm:FSTCaseConditionObjectType"/>
<xsd:element name="FSTCaseConditionObject" type="bpm:FSTCaseConditionObjectType"/>
<xsd:element name="FSTCaseConditionObject" type="bpm:FSTCaseConditionObjectType"/>
<xsd:element name="FSTCaseConditionObject" type="bpm:FSTCaseConditionObjectType"/>
<xsd:element name="FSTCaseConditionObject" type="bpm:FSTCaseConditionObjectType"/>
<xsd:element name="FSTCaseConditionObject" type="bpm:FSTCaseConditionObjectType"/>
<xsd:element name="FSTCaseConditionObject" type="bpm:FSTCaseConditionObjectType"/>
<xsd:element name="FSTCaseConditionObject" type="bpm:FSTCaseConditionObjectType"/>
<xsd:element name="FSTCaseConditionObject" type="bpm:FSTCaseConditionObjectType"/>
<xsd:element name="FSTCaseConditionObject" type="bpm:FSTCaseConditionObjectType"/>
<xsd:element name="FSTCaseConditionObject" type="bpm:FSTCaseConditionObjectType"/>
<xsd:element name="FSTCaseConditionObject" type="bpm:FSTCaseConditionObjectType"/>
<xsd:element name="FSTCaseConditionObject" type="bpm:FSTCaseConditionObjectType"/>
<xsd:element name="FSTCaseConditionObject" type="bpm:FSTCaseConditionObjectType"/>
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).
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**

<table>
<thead>
<tr>
<th>Field</th>
<th>Key</th>
<th>Data element</th>
<th>Data Type</th>
<th>Length</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>FREQUEST_ID</td>
<td></td>
<td></td>
<td>CHAR</td>
<td>20</td>
<td>Request ID</td>
</tr>
<tr>
<td>SOURCE_CLIENT</td>
<td></td>
<td></td>
<td>CLNR</td>
<td>3</td>
<td>Source Client</td>
</tr>
<tr>
<td>XML_VER</td>
<td></td>
<td></td>
<td>DEC</td>
<td>6</td>
<td>XML Version Number</td>
</tr>
<tr>
<td>xml</td>
<td></td>
<td></td>
<td>STRING</td>
<td>0</td>
<td>String representation of Original XML</td>
</tr>
<tr>
<td>TRANS_xml_VER</td>
<td></td>
<td></td>
<td>DEC</td>
<td>5</td>
<td>XML Version Number</td>
</tr>
<tr>
<td>TRANS_xml</td>
<td></td>
<td></td>
<td>STRING</td>
<td>0</td>
<td>String representation of Transformed XML</td>
</tr>
<tr>
<td>TRANS_MSG_xml</td>
<td></td>
<td></td>
<td>STRING</td>
<td>0</td>
<td>String representation of Transformation Message XML</td>
</tr>
</tbody>
</table>
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.

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 BRFplus XML is FDT.

```xml
```

**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.

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>XMLNS</td>
<td>The XML namespace to uniquely identify BRFplus elements is FDTNS and the URI used is <a href="http://sap.com/fdt/transport">http://sap.com/fdt/transport</a></td>
</tr>
<tr>
<td>Client</td>
<td>Client ID of the source system from where the XML has been generated</td>
</tr>
<tr>
<td>Date</td>
<td>Date on which the XML file has been generated</td>
</tr>
<tr>
<td>BRFplusInitialVersion</td>
<td>Shows the current XML version of the source system from which the XML</td>
</tr>
</tbody>
</table>
has been generated

<table>
<thead>
<tr>
<th><strong>BRFplusVersion</strong></th>
<th>Shows the XML version that was requested while exporting the BRFplus object</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>BRFplusSchema</strong></td>
<td>Shows the schema (internal or external) of the exported file</td>
</tr>
<tr>
<td><strong>SAPRelease</strong></td>
<td>Shows the current SAP release of the source system</td>
</tr>
<tr>
<td><strong>SourceExportReqID</strong></td>
<td>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</td>
</tr>
<tr>
<td><strong>SystemID</strong></td>
<td>Shows the system ID of the source system</td>
</tr>
<tr>
<td><strong>Time</strong></td>
<td>Shows the time at which the XML file has been generated</td>
</tr>
<tr>
<td><strong>User</strong></td>
<td>Name of the user (logged into the source system and exported the data) who generated the XML file</td>
</tr>
<tr>
<td><strong>AP</strong></td>
<td>Shows whether the XML has been generated from a Business ByDesign system or not.</td>
</tr>
</tbody>
</table>

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:

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

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.

```xml
<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>Emp1. 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:ADMIN_DATA>
```
The following sub-elements are used within the admin data element:

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>APPLICATION_ID</td>
<td>Stores the application ID to which the object belongs</td>
</tr>
<tr>
<td>ID</td>
<td>Stores the ID of the exported object</td>
</tr>
<tr>
<td>NAME</td>
<td>Stores the name of the object being exported</td>
</tr>
<tr>
<td>SYSTEM_OBJECT</td>
<td>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).</td>
</tr>
<tr>
<td>MASTERDATA_OBJECT</td>
<td>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).</td>
</tr>
<tr>
<td>CREATION_TIMESTAMP</td>
<td>Stores the timestamp at which the object has been created</td>
</tr>
<tr>
<td>CREATION_USER</td>
<td>Stores the name of the user who created the object which is being exported</td>
</tr>
<tr>
<td>CHANGE_TIMESTAMP</td>
<td>Stores the time at which the object has last been changed</td>
</tr>
<tr>
<td>VERSIONS</td>
<td>Stores the version details of the exported object</td>
</tr>
<tr>
<td>ACCESS_LEVEL</td>
<td>Stores the access level of the exported object</td>
</tr>
<tr>
<td>LOCAL</td>
<td>Indicates whether the exported object is a local object or not</td>
</tr>
<tr>
<td>TEXT_DEPENDENCY_TYPE</td>
<td>Stores the dependency type of the texts of an object.</td>
</tr>
<tr>
<td></td>
<td>The dependency can be:</td>
</tr>
<tr>
<td></td>
<td>• Independent of language and version</td>
</tr>
<tr>
<td></td>
<td>• Language-dependent but not version-dependent</td>
</tr>
<tr>
<td></td>
<td>• Version-dependent but not language-dependent</td>
</tr>
<tr>
<td></td>
<td>• Language-dependent and version-dependent</td>
</tr>
<tr>
<td>TEXTS</td>
<td>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.</td>
</tr>
</tbody>
</table>
## 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:
- 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:

<table>
<thead>
<tr>
<th>Report Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>FDT_XML_EXPORT</td>
<td>Exports BRFplus content to an XML file.</td>
</tr>
<tr>
<td>FDT_XML_IMPORT</td>
<td>Imports information about the BRFplus object content from an XML file document into the target system.</td>
</tr>
<tr>
<td>FDT_DEBUG_XML_EXPORT_IMPORT</td>
<td>Debugs and tracks changes done by the XML import.</td>
</tr>
</tbody>
</table>
**FDT_XML_EXPORT**

**Program to Export BRFplus 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
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**

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.
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.
XML Import Using the UI

1. In the Tool menu, choose XML Import.
   
   The XML Import page is displayed.

   ![XML Import Page](image)

   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.

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.

<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>EXPORT_XML</td>
<td>Exports BRFplus contents to an XML document</td>
</tr>
<tr>
<td>EXPORT_XML_APPLICATION</td>
<td>Exports XML for an application including all contained objects</td>
</tr>
<tr>
<td>IMPORT_XML</td>
<td>Imports an XML document.</td>
</tr>
</tbody>
</table>

**EXPORT_XML**

The input parameters for this method are:

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ITS_OBJECT_ID</td>
<td>Provides a table of the IDs of the objects which are to be exported</td>
</tr>
<tr>
<td>IV_DEEP (Optional)</td>
<td>Indicates whether subordinate named objects should be exported or not</td>
</tr>
<tr>
<td>IV_SCHEMA (Optional)</td>
<td>Specifies the XML schema type expected as output</td>
</tr>
<tr>
<td>IV_TIMESTAMP</td>
<td>Timestamp at which the BRFplus object’s active version is selected. If the timestamp is not provided, the latest active version is selected.</td>
</tr>
<tr>
<td>IV_XML_VERSION (Optional)</td>
<td>Used for downgrading. Specifies the XML version which is to be exported.</td>
</tr>
</tbody>
</table>
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:

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>EO_DOM_TREE</td>
<td>XML document in DOM object tree representation.</td>
</tr>
<tr>
<td>EV_STRING</td>
<td>XML document in string representation.</td>
</tr>
<tr>
<td>ET_MESSAGE</td>
<td>Processing messages for the export.</td>
</tr>
<tr>
<td>ETS_FAILURE</td>
<td>Provides a list of object IDs for which the export has failed</td>
</tr>
</tbody>
</table>

**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:

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

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:

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ETS_SUCCESS</td>
<td>Provides a sorted list of BRFplus object IDs which are imported successfully</td>
</tr>
<tr>
<td>ETS_FAILURE</td>
<td>Provides a sorted list of BRFplus object IDs which are not imported successfully</td>
</tr>
</tbody>
</table>

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.

<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>EXPORT_XML</td>
<td>Exports information about the BRFplus objects to the XML document</td>
</tr>
<tr>
<td>IMPORT_XML</td>
<td>Imports information about the BRFplus objects from the XML document</td>
</tr>
</tbody>
</table>
**EXPORT_XML**

The input parameters for this method are:

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>IV_TIMESTAMP</td>
<td>Specifies the active timestamp version</td>
</tr>
<tr>
<td>IO_PARENT</td>
<td>Describes the XML element instance in which the content needs to be appended</td>
</tr>
<tr>
<td>IO_NAMESPACE_ATTR (OPTIONAL)</td>
<td>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.</td>
</tr>
</tbody>
</table>

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 will be available in the *How to Create Custom Expression Types* document that is planned to be published later in 2011.

**History of XML Versions**

<table>
<thead>
<tr>
<th>BRFplus XML Version No</th>
<th>Changes</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.01</td>
<td>Initial Version</td>
</tr>
<tr>
<td>1.02</td>
<td>Addition of Development Package for Application</td>
</tr>
<tr>
<td>1.03</td>
<td>Addition of short texts</td>
</tr>
<tr>
<td></td>
<td>Addition of text dependency type</td>
</tr>
<tr>
<td></td>
<td>Addition of document dependency type</td>
</tr>
<tr>
<td>1.04</td>
<td>Addition of local objects</td>
</tr>
<tr>
<td>1.05</td>
<td>Change in position of local objects</td>
</tr>
</tbody>
</table>
| 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
Carsten Ziegler, BRFplus a Business Rule Engine written in ABAP
Carsten Ziegler, Important Information for Using BRFplus

Online Help
For more information, visit the Business Rules Management homepage.
Copyright

© Copyright 2011 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 Oracle Corporation.

JavaScript is a registered trademark of Oracle Corporation, 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.