Web Transaction API

Release 4.6C

Copyright

Icons

Icon	Meaning
⚠	Caution
💬	Example
💡	Note
🧭	Recommendation
syn	Syntax

Contents

Web Transaction API

Web Transaction API

The Web Transaction API is a set of ABAP macros and functions for programming Web transactions. You can use these macros and functions in ABAP programs to import data from, and export data to, the Internet Transaction Server (ITS).

Web Transaction Programming

Web transactions are Internet-enabled R/3 transactions that are initially developed as normal ABAP programs in the R/3 System, using the development tools of the ABAP Workbench.

When an Internet user starts a Web transaction, the ITS merges screen field contents returned by R/3 into a corresponding HTML template. The expanded HTML source is then sent to the user's Web browser and displayed as an HTML page.

The dialog programming model is the standard SAP programming model used in the R/3 System to develop transactions. This model is well established in the R/3 System kernel, but it lacks support for certain features required by Internet programming. To handle these features, the ITS offers a programming interface called the Web Transaction API.

Web Transaction API

The Web Transaction API supports the transfer of context fields and Multipurpose Internet Mail Extension (MIME) data between an R/3 transaction and a Web browser.

R/3 transactions written in ABAP can thus:

- Export context fields for use in HTML templates

  For example, an application could export a very long descriptive text for a material.

- Export arbitrary MIME types to the Web browser

  For example, an application could display an Excel spreadsheet taken from the SAP database in a Web browser.

- Import context fields from the ITS

  For example, data entered in an HTML tag `<TEXTAREA>` can be retrieved from the ITS and stored in an ABAP internal table for further processing.

- Import arbitrary MIME types from the Web browser

  For example, an application could use the HTML tag `<INPUT TYPE=FILE>` to upload a file from the workstation on which the Web browser is running into the SAP database.

- Import all context fields posted to ITS from the Web browser

  This function is useful if there is a large number of input fields, but it would not be practical to display them all on the screen.

- Redirect the Web browser to a URL, or post data to a URL

  These functions are useful for integrating external product catalogs.

For further information, see:

Preparing the Application [Page 8]

Processing Context Fields [Page 9]

Processing MIME Data [Page 15]

Controlling the Web Browser [Page 25]

Utility Functions [Page 28]

Preparing the Application

In order to use Internet Transaction Server (ITS) macros, you must include the module AVWRTCXM in the ABAP program.

```
PROGRAM SAPMPW02.
INCLUDE AVWRTCXM.
...
```

AVWRTCXM stands for **A**LE **v**ia **W**eb **R**untime **C**ontext **M**acros.

Processing Context Fields

An HTML template may contain expressions that refer to values on an R/3 screen. These values are part of the Internet Transaction Server (ITS) context, which includes the values on the current R/3 screen, as well as system information like the current Web browser type.

When you send data sent to the ITS via the Web Transaction API, the data becomes part of the ITS context, and you can access a context value by specifying the context field name in the macro.

The ITS context is maintained for a single dialog step only. After each dialog step, the context is purged and all values are reset. For the next dialog step, you must make the data available again using the ITS macros.

For information about processing context fields, see:

Processing an ITS Request [Page 10]

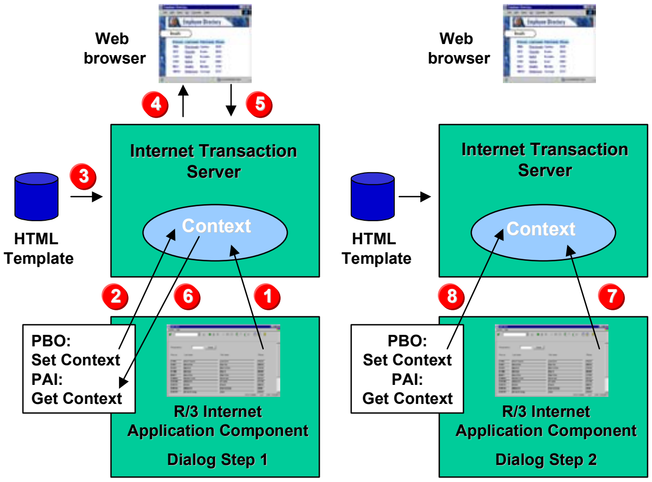
Macro FIELD-SET [Page 11]

Macro FIELD-TRANSPORT [Page 12]

Macro FIELD-GET [Page 13]

Processing an ITS Request

The following graphic shows how a request is processed when you use Internet Transaction Server (ITS) macros to import and export context fields:



1. Screen fields are automatically added to the ITS context.
2. If you need to set additional fields using the ITS macros, you do this in the PBO module.
3. After PBO, the ITS accesses the appropriate HTML template and merges the fields in the current ITS context into the template.
4. The final HTML page is then sent to the Web browser.
5. The user enters data and submits a request from the Web browser. The request fields are merged with the ITS context fields. If R/3 fields are present, the request fields are automatically entered into these fields. An OK code is sent to the transaction that initiates processing on the application server.
6. In the PAI module, the application can use ITS macros to import data from the ITS context.
7. After PAI in dialog step 1, the ITS context is reset and dialog step 2 starts.
8. Dialog step 2 continues.

Macro FIELD-SET

You use the FIELD-SET macro to set the value of context fields in the PBO module of a transaction. Multiple FIELD-SET calls are possible in separate PBO modules.

Usage

```
field-set   contextfield-name       (in,CHAR)
contextfield-index      (in,NUM)
contextfield-value      (in,CHAR)
```

Parameters

Parameter	Description
`Contextfield-name`	Name of the context field.
`contextfield-index`	Index of the context field.
`contextfield-value`	Value of the context field.

Example

```
field-set 'Company-Name' 1 'SAP Technogloy Inc'.
```

…

```
loop At Long_Text.
      field-set 'Long_Text' sy-tabix long_text-text.
Endloop.
```

An HTML template that uses these values may be like the following:

```
<HTML>
<HEAD> Internet Transaction Server Macros - Export Data </HEAD>
<H1> Internet Transaction Server Demo For `Company-Name` </H1>
`for textline in long_text`
`textline`
`end`
</HTML>
```

Macro FIELD-TRANSPORT

You use the FIELD-TRANSPORT macro to flush the buffer and transport all previously accumulated field assignments to the Internet Transaction Server (ITS). This is necessary, because FIELD-SET calls are not always transferred to the ITS immediately for performance reasons. All context fields assignments are first buffered on the application server.

Usage

```
field-transport.
```

Parameters

There are no parameters.

Example

```
field-set ‘Company-Name’      1 ‘SAP Technogloy Inc’.
field-set ‘Company-Street’    1 ‘950 Tower Lane’.
field-set ‘Company-City’      1 ‘Foster City’.
field-transport.
```

Macro FIELD-GET

You use the FIELD-GET macro to get the value of context fields in the PAI module of a transaction. FIELD-GET calls are not buffered, but executed immediately. Since this requires a separate request to the Internet Transaction Server (ITS) over the network, using this macro has performance implications. Therefore, you should use it only if you have to get mass data from the ITS context. Otherwise, use regular R/3 fields.

Usage

```
field-get   contextfield-name       (in,CHAR)
contextfield-index      (in,NUM)
contextfield-value      (out,ITAB)
contextfield-length     (out,NUM)
```

Parameters

Parameter	Description
`contextfield-name`	Name of the context field.
`contextfield-index`	Index of the context field.
`contextfield-value`	Value of the context field. The macro accepts any internal table and copies the value of the context field to the internal table components. If one internal table record is not enough, it adds additional records.
`contextfield-length`	Length of the context field

If the context field does not exist, this macro sets SY-SUBRC to a value other than zero. If execution completes successfully, the macro sets SY-SUBRC to 0.

Example

A common use for this function is to retrieve data from an HTML text area. In R/3, this HTML feature could be handled by a step loop, unless the text area contains a very large amount of data. In this case, the step loop may not be large enough to hold all the data coming from the Web browser, and text truncation may occur. An alternative is to use the FIELD-GET macro in the PAI module of the transaction.

Consider the following HTML code:

```
<HTML>
<HEAD> Internet Transaction Server Macros – Textarea </HEAD>
<H1> Please enter some data into this textarea </H1>
<FORM ACTION=″`wgateUrl(~OkCode=″SUBM″)`″ METHOD=″POST″>
<TEXTAREA NAME=″Textpad″ WRAP=″PHYSICAL″ ROWS=″10″ COLS=″40″
</TEXTAREA>
```

Macro FIELD-GET

```
</FORM>
</HTML>
```

The transaction could look like this:

```
Process Before Output:
…


Process After Input:

      Module Fcode.


Module Fcode.
      If Fcode = ‘SUBM’.
            Perform Get_Textarea.
      Endif.
Endmodule.
* ------------------------------------------------

Form Get_Textarea.
      Data: Datalen Type I.
      Data: Idx Type i.
      Data: Longtext(500) Occurs 0 With Header Line.

   Idx = 1.

   Loop.
            Field-Get ‘Textpad’ Idx Longtext1 Datalen.
            If Sy-Subrc <> 0.
                  Exit.
            Endif.
            Append Lines Of LONGText1 To LongText.
            Idx = Idx + 1.
      Endloop.
EndForm.
```

Processing MIME Data

For information about processing Multipurpose Internet Mail Extension (MIME) data, see:

Exporting MIME Data [Page 16]

Importing MIME Data [Page 17]

Macro MIME-DOWNLOAD [Page 18]

Macro MIME-TRANSPORT [Page 19]

Macro MIME-INFO [Page 20]

Macro MIME-UPLOAD [Page 22]

Macro MIME-UPLOADPART [Page 23]

Function ITS_IMPORT_CONTEXT [Page 24]

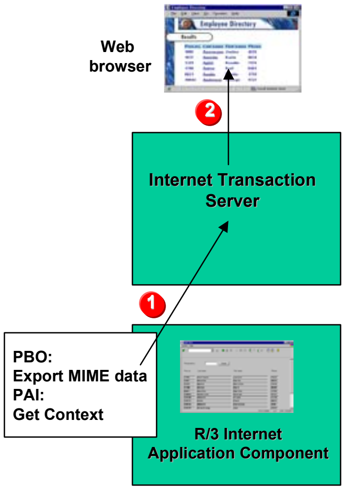
Exporting MIME Data

To generate HTML source code, the Internet Transaction Server (ITS) merges the context data into an HTML template. However, the Web browser can also handle other data types by using “helper” applications.

For example, a mail inbox Internet Application Component (IAC) could display all the items in the SAPoffice inbox on the Web. You can do this using HTML alone, but a mail could contain an attachment such as a Microsoft Excel spreadsheet.

The ITS is capable of sending arbitrary Multipurpose Internet Mail Extension (MIME) data, or MIME types to the Web browser. In the example described below, the IAC sends the Microsoft Excel attachment to the Web browser. Instead of using the `text/html` MIME type, the IAC uses `application/excel`. If the Web browser has a “helper” application configured to handle this MIME type, the browser automatically starts the “helper” and passes it all data received from the ITS.

The following graphic shows how the ITS processes a request that exports MIME data.


Web browser
2
Internet Transaction Server
1
PBO:
Export MIME data
PAI:
Get Context
R/3 Internet Application Component

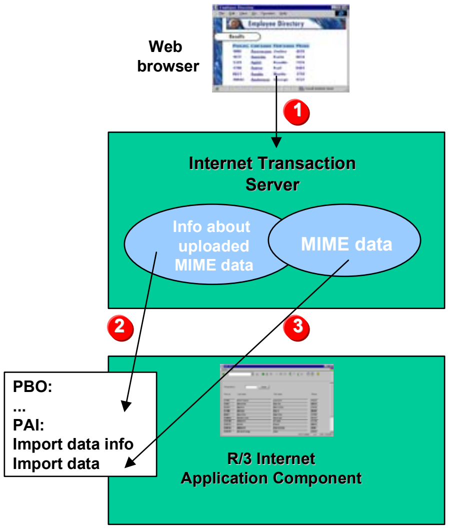
1. At PBO, the application can export arbitrary MIME data.
2. The ITS passes the received MIME data to the helper application started on the Web browser. If the transaction has exported MIME data for an OLE-enabled product, and the Microsoft Internet Explorer is being used, the helper application is started within the Web browser.

Importing MIME Data

Importing Multipurpose Internet Mail Extension (MIME) data is currently supported for Netscape Navigator 3.01 (or higher) and Microsoft Internet Explorer 3.01 (or higher). These browsers can handle a special HTML tag `<INPUT TYPE=FILE>`. They display a file selection box that allows users to select a file from the local file system. The file does not need to be an HTML file - it may contain any arbitrary MIME data.

The MIME data is uploaded to the Internet Transaction Server (ITS). At PAI, the Internet Application Component (IAC) can first retrieve information about the data uploaded to the ITS. It can then upload some or all of the actual data to the application server.

The following graphic shows how the ITS processes a request that imports MIME data.


Web browser
1
Internet Transaction Server
Info about uploaded MIME data
MIME data
2
3
PBO:
...
PAI:
Import data info
Import data
R/3 Internet Application Component

1. For every HTML form field of input type `File,` the Web browser offers a file selection box. The user can then select a local file that is then uploaded to the ITS.
2. The ITS holds information about all uploaded MIME data in its context. The IAC can then retrieve information about the MIME data first to select what needs to be uploaded to the application server.
3. Some or all MIME data collected by the ITS is uploaded to the application server.

Macro MIME-DOWNLOAD

You use the MIME-DOWNLOAD macro to transfer arbitrary Multipurpose Internet Mail Extension (MIME) data to the Internet Transaction Server (ITS) in the PBO module of a transaction. The ITS then sends the MIME data to the Web browser without generating an HTML template. Only one MIME-DOWNLOAD is possible during PBO processing.

Usage

```
mime-Download     Mimedata    (in,ITAB)
mimedata_Length   (in,NUM)
mimetype    (in,CHAR)
```

Parameters

Parameter	Description
mimedata	Internal table with arbitrary structure. The MIME-DOWNLOAD macro iterates over all internal table rows and sends the contents of each internal table component as part of the MIME data to the Web browser.
mimedata_length	Length of the data to be sent to the Web browser.
mimetype	MIME type.

Example

```
mime-download mimetab mimelength 'application/excel'.
```

Macro MIME-TRANSPORT

The MIME-TRANSPORT macro is fully compatible with the MIME-DOWNLOAD macro and has been kept for downward compatibility. See Macro MIME-DOWNLOAD [Page 18].

Macro MIME-INFO

You use the MIME-INFO macro to retrieve information about the data uploaded to the Internet Transaction Server (ITS).

Usage

```
mime-info   mimeinfo    (out,savwmpmime)
```

Parameters

Parameter	Description
`mimeinfo`	Internal table with information about MIME data temporarily stored in the ITS.

Example

This is an example of an HTML template that allows files to be uploaded from the PC where the Web browser is running. The special **`enctype`** tag must be present to ensure proper transfer from the Web browser to the ITS.

```
<HTML>
<FORM enctype="multipart/form-data"
ACTION="`wgateUrl()`"
METHOD="POST">
      <INPUT      TYPE="FILE"       NAME="LOCALFILE1">
      <INPUT      TYPE="FILE"       NAME="LOCALFILE2">
      <INPUT      TYPE="SUBMIT"     NAME="~OkCode(FIUP)"
VALUE="File Upload">
</FORM>
</HTML>
```

The following is a code fragment for retrieving information about the uploaded files.

```
data: Mime_Info Like savwmpmime Occurs 0 With Header Line.
mime-info mime_info.
loop At mime_info.
* mime_info-conttype contains the MIME type
* mime_info-contlength contains the length
* mime_info-contname contains the original HTML field name
* mime_info-contfname contains the file name on the browser pc
endloop.
```

The following table shows possible contents of the MIME_INFO table:

conttype	contlength	contname	contfname
Application/excel	12345	LOCALFILE1	C:\TEMP\LOCALFILE1.XLS
Application/ octet-stream	54321	LOCALFILE2	C:\TEMP\LOCALFILE2.DOC

If the MIME type is unknown to the Web browser, MIME_INFO-CONTTYPE will contain `Application/octet-stream.`

Macro MIME-UPLOAD

You use the MIME-UPLOAD macro to upload Multipurpose Internet Mail Extension (MIME) data from the Internet Transaction Server (ITS) to the application server.

Usage

```
mime-upload mimeinfo_index    (in,NUM)
mimedata    (out,ITAB)
mimedata_length   (out,NUM)
```

Parameters

Parameter	Description
MIMEINFO_INDEX	Index into the MIME-INFO ındex.
MIMEDATA	Internal table into which the data is copied. The macro accepts any internal table and copies the value of the context field to the internal table components. If one record is not sufficient, it adds additional records.
MIMEDATA_LENGTH	Length of the retrieved data.

Example

The following is a complete example of how to upload MIME data smaller than 100 KB from the ITS to the application server:

```
data: mime_info like savwmpmime occurs 0 with header line.
data: datalen type i.
data: data_tab(500) occurs 0 with header line.


mime-info mime_info.
loop at mime_Info.
      If mime_info-contlength < 10000.
            mime-upload sy-tabix data_tab datalen.
      endif.
endloop.
```

Macro MIME-UPLOADPART

You use the MIME-UPLOADPART macro to upload a specific section of Multipurpose Internet Mail Extension (MIME) data from the Internet Transaction Server (ITS) to the application server. The macro examines the contents of very large files before uploading the complete file to the application server, and allows you to read and process MIME data piece by piece, rather than reading everything into one huge internal table and then processing the data.

Usage

```
mime-upload mimeinfo_index    (in,NUM)
mimedata_offset   (in,NUM)
mimedata_length   (in,NUM)
mimedata     (out,ITAB)
```

Parameters

Parameter	Description
MIMEINFO_INDEX	Index into the MIME-INFO index
MIMEINFO_OFFSET	Start position of part of MIME data
MIMEINFO_LENGTH	Length of part of MIME data
MIMEDATA	Internal table into which the data is copied. The macro accepts any internal table and copies the value of the context field to the internal table components. If one record is not sufficient, it adds additional records.

Function ITS_IMPORT_CONTEXT

This function is useful if the number of input fields required to transfer all Web input does not fit on one screen, or if the number of input fields is unknown. The functions returns all context fields from the Internet Transaction Server (ITS) to an internal table.

Usage

```
its_import_context     tables     context
```

Parameters

Parameter	Description
CONTEXT	Table of structure SAVWCTXT.

CONTEXT has the following members:

Member	Description
FIELDNAME	Name of the context field.
FIELDINDEX	Index of the context field, starting with 1.
FIELDCONT	Content of the context field.

As the size of FIELDCONT is limited, the value of the context field is truncated if it is longer than 255 bytes. Use FIELD-GET if you need to transfer larger amounts of data.

Example

The following example shows how to process all input fields transferred from the Web.

```
FORM PROCESS_WEB_INPUT.

   DATA: CONTEXT LIKE SAVWCTXT OCCURS 0 WITH HEADER LINE.

   CALL FUNCTION 'ITS_IMPORT_CONTEXT'
      TABLES
         CONTEXT = CONTEXT
      EXCEPTIONS
         ITS_NOT_AVAILABLE = 1
         OTHERS            = 2.

   IF SY-SUBRC <> 0.
      EXIT.
   ENDIF.

   LOOP AT CONTEXT.
  ...
   ENDLOOP.

ENDFORM.
```

Controlling the Web Browser

For information on Web browser control macros, see:

Macro ITS_BROWSER_REDIRECT [Page 26]

Macro ITS-BROWSER_POST [Page 27]

Macro ITS_BROWSER_REDIRECT

You use this macro to redirect the Web browser to a specific URL. An application can use this macro to point the Web browser to a URL rather than processing an HTML template.

Usage

```
its-browser_redirect          Url    (in,CHAR)
```

Parameters

Parameter	Description
URL	URL to which the application would like the Web browser to point.

Example

```
form redirect_browser.
   its-browser_redirect ‘http://www.saptech.com’.
endform.
```

Macro ITS-BROWSER_POST

You use this macro to post data to a specific URL. You could, for example, use it in an application to post a product catalog to external Web sites. You specify the URL to which the application is to post data as a parameter, but all parameters that the application is to post to the Web site must be specified with previous FIELD-SET calls.

Usage

```
its-browser_post        Url    (in,CHAR)
```

Parameters

Parameter	Description
URL	URL of the Web site.

This macro creates a HTML page dynamically with one HTML FORM and one hidden field per POST parameter. The JavaScript function `submit()` automatically posts the data to the URL. Please note that the function works only in Web browsers enabled for JavaScript 1.0. If the data cannot be posted to the URL, the user gets an error message in the Web browser window.

Example

```
form post_to_catalog.

   field-set ‘Quantity’ 1 ‘10’.
   field-set ‘Material’ 1 ‘Material_1’.
   field-set ‘Quantity’ 2 ‘20’.
   field-set ‘Material’ 2 ‘Material_2’.
   its-browser_post ‘http://www.sap.com/cgi-bin/catalog.exe’.

endform.
```

Utility Functions

For information on utility functions, see:

Function ITS_PING [Page 29]

Function ITS_PING

You use this function to check whether the transaction is called by the Internet Transaction Server (ITS) or in dialog mode.

Usage

```
its_ping
```

The function raises the exception ITS_NOT_AVAILABLE if the calling transaction is called in dialog mode.

Example

```
form check_its.

   call function ‘ITS_PING’
      exceptions
         its_not_available = 1
         others            = 2
   if sy-subrc = 0.
*    transaction called by its
   endif.

endform.
```