



These notes contain the following:

- ✦ New features
- ✦ Changes in behavior
- ✦ Problems fixed
- ✦ Deprecated

New features

The Skype API 3.0 introduces the following features:

- ✦ Interfaces to support the Extras Framework for Skype
- ✦ Call transfer interface
- ✦ New commands to personalize Skype



Interfaces for the extras framework

Skype 3.0 introduces a number of interfaces to enable the Extras Framework. The following section describes these interfaces.

Important: Although you can use these interfaces to create customized menus and events for your Extra, Skype recommends that you enter menu and event information in the appropriate fields in Publishing Studio, to provide tighter integration with the Extras Framework.

Menu interface

The `MENU` interface provides the commands, notifications, and events required to create and manage menu entries in the Skype client. Each API client has its own specific menu items and each client "sees" and receives events only from menus it creates. Menu items are removed if the API client disconnects.

Menu item properties

When you create a menu item, you must define mandatory properties and you can also define a number of optional properties.

Mandatory properties

It is mandatory to set the following properties for each menu item.

- ♦ The `id` is a unique alphanumeric identifier which must start with a letter.
- ♦ `CONTEXT`
Create a separate menu item for each context your application requires in the Skype client. Each menu item must define one Skype context:
 - `chat`
 - `call`
 - `contact`
 - `myself`
 - `tools`
- ♦ The `CAPTION` property is a mandatory property, containing the first line of the menu entry (appears in black font). Maximum 32 characters. Enclose in quotes if the text contains whitespaces.



Optional properties

You can also set a range of optional properties.

Table 1: Optional properties for menu items

Property	Description
HINT <hint>	Text string which contains the second line of the menu entry (appears in gray font). Maximum 32 characters. Enclose in quotes if the text includes whitespaces.
ICON <path>	The directory path to the .ico file of the menu icon. Maximum size 32 x 32 pixels. Enclose in quotes.
ENABLED {true false}	Indicates whether to display the menu item at start up. <ul style="list-style-type: none"> • true = display • false = don't display
CONTACT_TYPE_FILTER <all skypeid skypeout>	Relevant only when the CONTEXT is contact, to filter contacts by: <ul style="list-style-type: none"> • all contacts • SkypeName • SkypeOut contacts
ENABLE_MULTIPLE_CONTACTS <true false>	If false, the menu item is disabled in group chats, conference calls, and is not enabled for user multi-selection.

Menu item commands

Skype recommends that you define menu items in Publishing Studio. However, you can use the following commands if you want to manage menu items independently:

- ✦ CREATE MENU_ITEM
- ✦ DELETE MENU_ITEM
- ✦ SET MENU_ITEM

CREATE MENU_ITEM

To create a MENU_ITEM you must define properties in the correct order:

- ✦ id
- ✦ Skype CONTEXT
- ✦ plug-in CAPTION
- ✦ any optional properties in the precise order they appear in Table 1 on page 3.



Syntax

```
-> CREATE MENU_ITEM ab1 CONTEXT  
(<chat|call|contact|myself|tools>) CAPTION "text menu 1"
```

Response

```
<- MENU_ITEM ab1 CREATED
```

Example

```
-> CREATE MENU_ITEM ab2 CONTEXT chat CAPTION "text menu 1" HINT  
"text menu 2" ICON "C:\WINDOWS\Zapotec.bmp"  
  
<- MENU_ITEM ab2 CREATED
```

DELETE MENU_ITEM

To delete an entry, use the DELETE MENU_ITEM command.

Syntax

```
-> DELETE MENU_ITEM ab1
```

Response

```
<- DELETE MENU_ITEM ab1
```

SET MENU_ITEM

You can modify the following menu item settings:

- ✦ CAPTION
- ✦ HINT
- ✦ ENABLED

You cannot modify any of these properties in the same SET command. You must use a separate SET command to change each individual property.

Syntax

```
-> SET MENU_ITEM ab1 CAPTION "menu text 1"
```

Response

```
<- MENU_ITEM ab1 CAPTION "menu text 1"
```

MENU_ITEM event

The client sends a notification event when a user clicks a menu item.

```
<- MENU_ITEM <id> CLICKED [<userid>[, <userid>]*] CONTEXT  
<chat|call|contact|myself|tools> [CONTEXT_ID <id-list>]
```



Events interface

All events are specific to the API client that created them. Only the client which creates an event, can receive and delete events. Events are removed when the API client disconnects.

Event properties

An `EVENT` has the following properties:

- ✦ `id`
- ✦ `CAPTION`
- ✦ `HINT`

Event commands

The plug-in API introduces the following commands for managing events:

- ✦ `CREATE EVENT`
- ✦ `DELETE EVENT`

CREATE EVENT

To create an `EVENT`:

Syntax

```
-> CREATE EVENT <id> CAPTION <text> HINT <text>
```



Call transfer interface

With the call transfer API, users can transfer calls to other users, for example:

A calls B and B transfers the call to C, enabling a call between A and C.

To enable call transfer, Skype 3.0 introduces a new protocol as well as new status values, properties, and commands.

Protocol

Skype 3.0 introduces protocol 7 to support call transfer.

Call transfer status values

Skype 3.0 introduces the following status values for the `CALL` object:

- ✦ The `TRANSFERRING` status is set for caller B while the call between A and C continues.
- ✦ The `TRANSFERRED` status is set for caller B either when the transferred call ends or when caller B sends an `END` or `HANGUP` command. This value remains in the history for the call.

Call transfer properties

Skype 3.0 introduces the following properties for the `CALL` object:

- ✦ The `TRANSFER_STATUS` is sent continuously to caller B by caller A until the call ends or until caller B sends the `CALL ALTER END` command. This command will end the call for caller B but will not terminate the call between A and C, just the status updates.
- ✦ The `TRANSFER_ACTIVE` property is seen by caller A during the call.
- ✦ The `TRANSFERRED_BY` property enables caller C to see who transferred the call.

Call transfer commands

Skype 3.0 introduces a number of commands to support call transfer.

GET CALL CAN_TRANSFER

Use the `GET CALL CAN_TRANSFER` command to query if a user can transfer a call.

Syntax

```
-> GET CALL <id> CAN_TRANSFER <Skypename|number>
```

Response

```
<- CALL <id> CAN_TRANSFER <Skypename|number> <TRUE|FALSE>
```



ALTER CALL TRANSFER

Use the ALTER CALL TRANSFER command to transfer a call to a group. The first member of the group to answer will get the call.

Syntax

```
-> ALTER CALL <id> TRANSFER handle1[, handle2 ..]
```

Response

```
<- ALTER CALL <id> TRANSFER
```

Call transfer errors

Error 592 – ALTER CALL: transferring the call failed

Error 593 – GET CALL: CAN_TRANSFER invalid target



Personalize commands

Skype 3.0 introduces the following commands to enable the user to control the appearance of the Skype client.

- ✦ GET WALLPAPER
- ✦ SET WALLPAPER

GET WALLPAPER

Use the GET WALLPAPER command to query the image used for the background appearance of the client.

Syntax

```
-> GET WALLPAPER
```

Response

```
<- WALLPAPER <file.bmp>
```

The image file can be any of the following file formats:

- ✦ .png
- ✦ .jpg
- ✦ .bmp

SET WALLPAPER

Use the SET WALLPAPER command to change the image displayed in the client.

Syntax

```
-> SET WALLPAPER <file.bmp>
```

Response

```
<- WALLPAPER <file.bmp>
```



Changes in behavior

Skype 3.0 introduces a change in text parsing. Use double quotes for all text strings that contain whitespaces. For example:

```
SET MENU_ITEM a1 SET CAPTION "My cool plug-in"  
SET MENU_ITEM a2 SET CAPTION Coolplug-in
```

If you omit quotes around a text string that includes whitespace, all information after the first whitespace is ignored. For example:

```
SET MENU_ITEM a1 SET CAPTION My cool plug-in  
sets the menu item as My
```

Use double quotes for all commands that pass a comma-separated list to Skype, such:

- ✦ ALTER CHAT ADDMEMBERS
- ✦ CHAT CREATE
- ✦ CREATE SMS
- ✦ SET SMS TARGET_NUMBERS

Example syntax

```
CALL "12345", "4567", 45678, "345678", "abc, def", "invalid user"
```

- ✦ CALL "test, test_2" – calls a user test, test_2
- ✦ CALL "test, test_2" – ERROR 92 CALL: Unrecognised identity (" cannot be in username). Note there is a space after the comma separator in this string.
- ✦ CALL test, test_2 – initiates a conference to users test and test_2
- ✦ CALL test, test_2 – initiates a conference to test and test_2