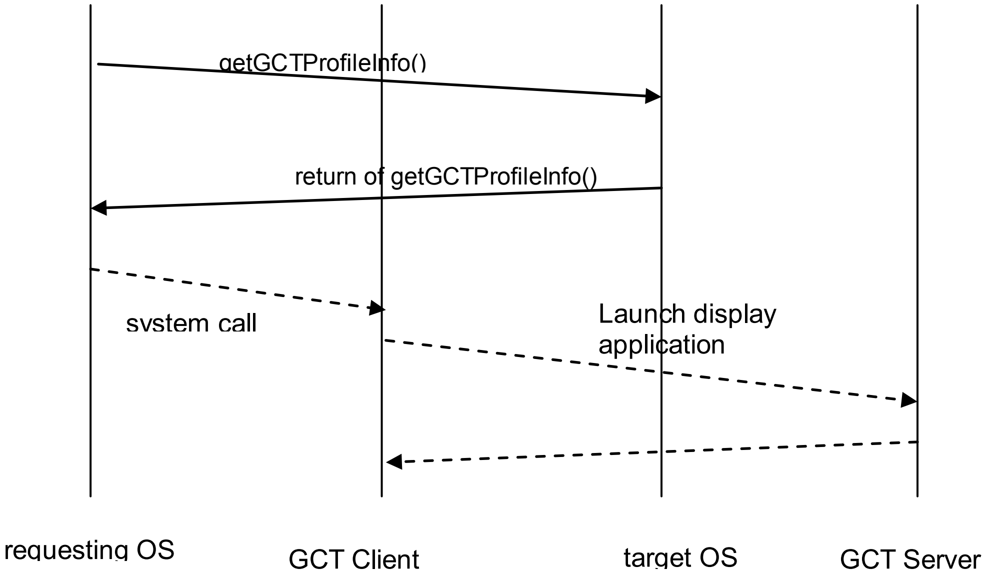
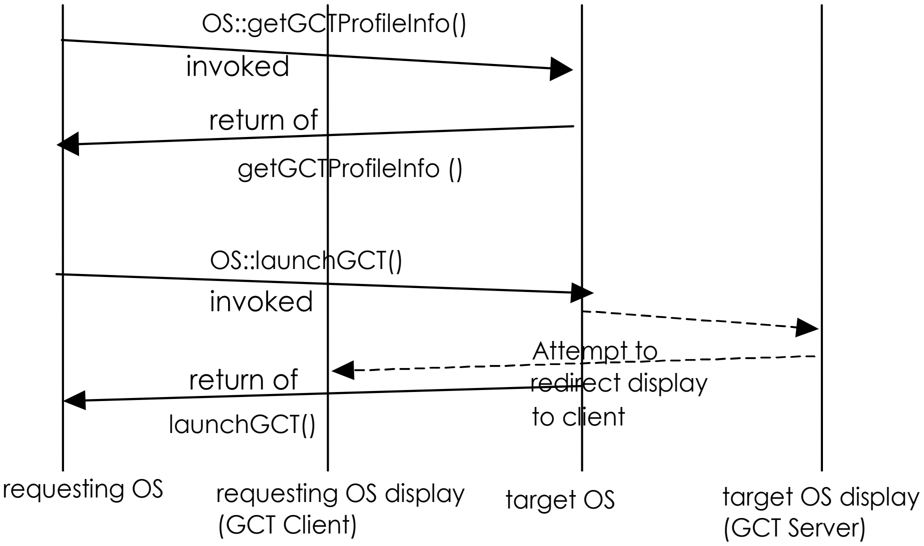
GUI Cut-Through



Client-Launch Configuration Sequence



Server Launch Sequence

Note that it is not required for the requesting OS to re-obtain the GCT profile info. before every launchGCT request

1 Conventions for the Client-Launch

The following placeholders are used whenever an target OS command requires specific information from the requesting OS. These placeholders are used as required by the target OS to retrieve the object name, such as ManagedElement required for ME scope GUI Cut-Through windows. The following conventions are used (and should be noted in the IDL solution set):

<OS>	the value of "target OS" from the NameAttribute name of the target OS
<ManagedElement>	the value of "ManagedElement" from the NameAttribute name of the ME

The minimal scope that shall be implemented is the target OS scope. The only mandatory window is the target OS top level window. The recommended implementation would include the ME scope. However, there is no limit of scoping that is mandated by the interface. The requesting OS implementation shall ignore GCT scopes that are supported by the target OS which it does not require.

<GCTDisplay>	Address for GCT display
<userId>	userid
<password>	password associated with the userid; may be encrypted with pairwise agreement between requesting OS and target OS
<capability>	capability that is assigned to the given user within the GCT application. Suggested values are: "READ_WRITE", "READ_ONLY"
\>	used to escape the > character
\<	used to escape the < character
\\	used to escape the \ character

Additional parameters can be added as required such as resolution data. These additional parametersrequire bilateral agreement between the target OS and the requesting OS. For example to extend the scope to objects beyond the target OS and ME object the following templates might be used: <PTP> , <CTP> or <Equipment>

requesting OS launches the GCT by executing the command line with the placeholders replaced with the corresponding data.

For example, for the following "command" value that is returned from the getGCTProfileInfo operation:

"/opt/bin/invoke_GCT -display <GCTDisplay> -desc targetOSTop –os <target OS>"

The expanded command that is launched could be:

```
/opt/bin/invoke_GCT -display 2.3.4.5:0.0 -desc targetOSTop -os mytargetOS
```

and, if the requesting OS were using remote shell on a Unix platform (based on the emsGCTPlatform value returned in the getGCTProfileInfo), it would execute the following:

```
remsh  200.300.400.500  -n "export DISPLAY=2.3.4.5:0.0; /opt/bin/invoke_GCT -
display 2.3.4.5:0.0 -desc targetOSTop -os mytargetOS"
```

where “200.300.400.500” is the emsGCTHostname returned in the getGCTProfileInfo.

The launch mechanism itself and security issues are addressed outside of this interface.

There is no error returned from the GCT command. This approach provides consistency for all commands executed since not all commands can return errors.

2 Administrative Appendix

2.1 Document History

Version	Date	Description of Change
3.0	November 2006	Conversion to new template.

2.2 Acknowledgments

First Name	Last Name	Company

2.3 How to comment on this document

Comments and requests for information must be in written form and addressed to the contact identified below:

Keith	Dorking	CIENA
Phone:	+1 678 867 5007	
Fax:	+1 678 867 5010	
e-mail:	Kdorking@ciena.com	

Please be specific, since your comments will be dealt with by the team evaluating numerous inputs and trying to produce a single text. Thus we appreciate significant specific input. We are looking for more input than wordsmith" items, however editing and structural help are greatly appreciated where better clarity is the result.